SOJOURN DISTRIBUTIONS FOR PARTICULAR CUSTOMERS IN NETWORKS OF QUEUES

By
RUSSELL EDWARD KING

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA
1986

Copyright 1986
By
Russell Edward King

This dissertation is dedicated to
my parents, Doris and Maxwell,
and loving wife, Christina

ACKNOWLEDGMENTS

I wish to acknowledge the considerable efforts of Dr. Thom J. Hodgson who not only served as chairman of my master's and doctoral committees, providing inspiration and motivation through the trying times, but who also helped in finding the environment to make the pursuit of this degree economically feasible. His guidance and instruction both on and off the handball court helped the shape and form of this research, as well as, my personal and career development.

I would also like to thank the other members of my committee, Dr. Louis Martin-Vega, Dr. Ralph Swain, Dr. Kerry Kilpatrick, and Dr. James McClave for their comments and suggestions concerning this research. I am also indebted to Dr. Donald Hearn and Dr. Jack Elzinga for their help in providing the opportunity to begin my teaching career. Additionally, I would like to express my thanks to Dr. Louis Martin-Vega for the hours of talk and guidance outside the research.

Finally, I would like to express my deepest thanks to my family and friends for their constant support. In particular, I wish to point out the people to whom this dissertation is dedicated. First, my mother, Doris W.

King, for the love and instillation of the desire to learn. To my father, Dr. Maxwell C. King, the ultimate self-made man, who gave me the love of sports and the desire to excel, and who was from the start and will always be, my idol and ideal. And, finally, to my dearest wife Christina who endured the sacrifices in the early years of our marriage and continues to love me just the same.

This dissertation was supported in part by a grant from the Office of Naval Research.

TABLE OF CONTENTS

ACKNOWLEGEMENTS iv

LIST OF FIGURES xi

LIST OF TABLES xii

ABSTRACT xv

CHAPTERS

1 INTRODUCTION 1

1.1 Purpose 1
1.2 Motivation 3
1.3 Scope 8
1.4 Overview 9

2 LITERATURE REVIEW 11

2.1 Introduction 11
2.2 Network Queueing Review 11
2.3 Sojourn-Time Results Review 22

3 SERIAL QUEUEING MODELS 33

3.1 Introduction 33
3.2 Exact Analytic Solution 36
3.2.1 Problem Specification 36
3.2.2 Solution Procedure 38
3.2.3 A Small Example 45
3.2.4 Computational Considerations 57
3.3 Heuristic Solution Procedures 60
3.3.1 Monte-Carlo Simulation 60
3.3.2 Deterministic Simulation Heuristic (DSH) 61
3.3.3 Enhanced Deterministic Simulation Heuristic (DPL) 67
3.4 Experimental Design 74
3.5 Results 80
3.5.1 Computational Requirements of Exact Solution . . . 81
3.5.2 Monte-Carlo Simulation Validation and Calibration 82
3.5.3 Results of Heuristics 85
3.6 Conclusions 97

4 DISASSEMBLY-REASSEMBLY NETWORKS 99

4.1 Introduction 99
4.2 Model Description 100
4.3 Heuristic Solution Procedures 110
4.3.1 Deterministic Simulation Heuristic (DSH) 111
4.3.2 Enhanced Deterministic Simulation Heuristic (DPL) 111
4.3.3 Deterministic Simulation Heuristic with Subroute Maximization (DSHSM) 112
4.3.4 Enhanced Deterministic Simulation Heuristic with Subroute Maximization (DPLSM) 117
4.4 Experimental Design 117
4.5 Results 120
4.5.1 Comparison of Means 122
4.5.2 Comparison of Standard Deviations 127
4.6 Conclusions 132

5 SUMMARY AND SUGGESTIONS FOR FUTURE RESEARCH 134

5.1 Summary 134
5.2 Suggestions for Future Work 137

APPENDICES

A MISCELLANEOUS 142

Proof that the Minimum of M Exponentially Distributed Holding Times is Exponentially Distributed 142

Derivation of T_i^* for the Example of Section 3.2.3 . . 144

B LISTINGS OF PROGRAMS FOR HEURISTICS 148

Listing of the DSH Program 148
Listing of the DPL Program 151
Listing of the DSHSM Program 156
Listing of the DPLSM Program 160

C PERCENTAGE DIFFERENCES OF SELECTED EXAMPLES FOR SERIAL NETWORKS 167

Percentage Difference Between the MEANS of the EXACT Analytic Solution and the DSH and DPL Heuristics for the 2 Queue Model 1 Examples 167

Percentage Difference Between the MEANS of the MONTE CARLO Simulation and the DSH and DPL Heuristics for the 10 Queue Model 11 Examples . . . 172

Percentage Difference Between the MEANS of the MONTE CARLO Simulation and the DSH and DPL Heuristics for the 5 Queue Model 41 Examples . . . 175

Percentage Difference Between the STANDARD DEVIATION of the EXACT Analytic Solution and the DSH and DPL Heuristics for the 2 Queue Model 1 Examples 178

Percentage Difference Between the STANDARD DEVIATION of the MONTE CARLO Simulation and the DSH and DPL Heuristics for the 10 Queue Model 11 Examples 183

Percentage Difference Between the STANDARD DEVIATION of the MONTE CARLO Simulation and the DSH and DPL Heuristics for the 5 Queue Model 41 Examples 186

D PERCENTAGE DIFFERENCES OF SELECTED EXAMPLES FOR DISASSEMBLY-REASSEMBLY NETWORKS 190

Percentage Difference Between the MEANS of the MONTE CARLO Simulation and the DSH, DPL, DSHSM and DPLSM Heuristics for the 4 Queue Model 25 Examples 190

Percentage Difference Between the MEANS of the MONTE CARLO Simulation and the DSH, DPL, DSHSM and DPLSM Heuristics for the 8 Queue Model 45 Examples 197

Percentage Difference Between the STANDARD DEVIATION of the MONTE CARLO Simulation and the DSH, DPL, DSHSM and DPLSM Heuristics for the 4 Queue Model 25 Examples 201

Percentage Difference Between the STANDARD DEVIATION of the MONTE CARLO Simulation and the DSH, DPL, DSHSM and DPLSM Heuristics for the 8 Queue Model 45 Examples 208

BIBLIOGRAPHY 212

BIOGRAPHICAL SKETCH 218

LIST OF FIGURES

1.1 Routing Network for a Small Component Part 4

2.1 M/M/1 Queue with Feedback 26

2.2 Network with Overtaking 26

3.1 Serial Network of Single-Server Queues 36

3.2 State Transition Diagram 47

4.1 Serial Network of Single-Server Queues 101

4.2 Tandem Network of Single-Server Production Stages 103

5.1 Generic Serial Queueing Network 139

LIST OF TABLES

3.1 Example Problem State Labels 49

3.2 One Step Transition Matrix, T 52

3.3 Modified One Step Transition Matrix, T^* 53

3.4 Number of Paths in Some Representative Scenarios 59

3.5 Coefficient of Skewness for some Representative Distributions 68

3.6 Example Numbers for 2 Queue Scenarios 76

3.7 Example Numbers for 3 Queue Scenarios 77

3.8 Example Numbers for 5 Queue Scenarios 78

3.9 Example Numbers for 10 Queue Scenarios 79

3.10 Summary of Percentage Differences of the Estimated Sojourn Means from the DSH and DPL Techniques vs. the Exact Analytic Solution 87

3.11 Average Absolute Percent Difference for the Mean 88

3.12 Average Percent Difference for the Mean 88

3.13 Standard Deviation of the Average Percent Difference for the Mean 91

3.14 Absolute Percent Difference in the Mean for DSH for Some Particular Examples 91

3.15 Summary of Percentage Differences of the Estimated Sojourn Standard Deviations from the DSH and DPL Techniques vs. the Exact Analytic Solution 93

3.16 Average Absolute Percent Difference for the Standard Deviation 93

3.17 Average Percent Difference for the Standard Deviation . . . 94

3.18 Standard Deviation of the Average Percent Difference for the Standard Deviation 94

3.19 Absolute Percent Difference in the Standard Deviation for DSH for Some Particular Examples 97

4.1 Mean of the MAX$[T_1, T_2]$ 107

4.2 Variance of the MAX$[T_1, T_2]$ 108

4.3 Mean and Variance of Some Representative T_i 109

4.4 Example Numbers for 4 Queue Scenarios 119

4.5 Example Numbers for 8 Queue Scenarios 121

4.6 Average Absolute Difference for the Mean 123

4.7 Average Difference for the Mean 124

4.8 Standard Deviation of the Average Difference for the Mean 125

4.9 Average Absolute Difference for the Standard Deviation . . 128

4.10 Average Difference for the Standard Deviation 129

4.11 Standard Deviation of the Average Difference for the Standard Deviation 130

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of
the Requirements for the Degree of Doctor of Philosophy

SOJOURN DISTRIBUTIONS FOR PARTICULAR CUSTOMERS IN NETWORKS OF QUEUES

By

Russell Edward King

December, 1986

Chairman: Thom J. Hodgson
Major Department: Industrial and Systems Engineering

The purpose of this research is to study the transient behavior of sojourn distributions of particular customers traversing networks of single-server queues. This vein of research is motivated by the need to project completion times of critical customers in a loaded, capacitated queueing system. In particular, two different systems are considered, serial networks with First-In-First-Out queueing discipline and disassembly-reassembly networks which consist of serial networks with common beginning and ending queues. An

extensive literature search is included which reveals a lack of significant effort in this area.

An analytic model based on a Markovian state space is developed to determine to the sojourn distributions. The exact distribution is shown to be the so-called generalized hyper-Erlang. However, the solution of the model is shown to be computationally infeasible even for relatively small scenarios as the number of paths through the state space (and thus the number of terms in the distribution) explodes.

Given the failure of the exact solution, heuristic methodologies based on the Central Limit Theorem and observation of the behavior of the exact solution are developed as an alternative to digital Monte-Carlo simulation. These heuristics are shown to be accurate in approximating the mean and variance of the sojourn distribution in a variety of application scenarios. The heuristics are also shown to require minimal computer storage and CPU resources. As a result the heuristics are applicable to large-scale, real and near-real time applications.

CHAPTER 1

INTRODUCTION

1.1 Purpose

Queueing networks have received a great deal of attention in the literature ever since J.R. Jackson's first paper on the topic [33] appeared in 1957. This interest has been sparked by not only the wide variety of intriguing theoretical aspects of the analysis of such problems, but also a seemingly boundless array of real problems which can be modeled as such. Recent advances in computer, communication, and manufacturing technology only seem to ensure that research in this area will continue to remain fruitful and worthwhile.

The purpose of this research is to address two, as yet, largely unstudied aspects of queueing network problems. One aspect, that is of particular interest in a host of "real-world" applications, is the sojourn-time distribution for particular jobs traversing some network of queueing stations.

While much work has been done in determining such distributions under "steady-state" or equilibrium conditions, the same cannot be said for "given scenario" conditions. A "given scenario" is a given state of the system at a particular instant in time, i.e., the load at each queue, the status of each server, etc.

The other aspect to be addressed is a particular type of job that may be disassembled into component sub-jobs and later reassembled. In general, the disassembly and reassembly may occur at several levels. The generation of the sub-jobs and later reformation of the parent can be defined by a project network similar to a CPM/PERT network. This type of job is common to such things as maintenance and repair facilities, communications networks, and computer systems where parallel processing may occur.

The emphasis of this dissertation will be the presentation and evaluation of techniques for determining sojourn-time distributions for particular jobs in a network of queues where jobs may require disassembly and reassembly. Analytic techniques will be developed and shown to be impractical computationally for even small problems. Therefore, particular interest will be paid to simple heuristics that can provide fast, accurate projections of job

completion times under "given scenario" conditions that potentially can be used in real or near-real time applications.

1.2 Motivation

The motivation for studying queueing systems of this type stems from production problems encountered by management of repair and maintenance job shops, like Naval Air Rework Facilities (NARFs).

A NARF operates in a large job shop environment where planned, periodic depot level maintenance, (i.e., overhaul), of aircraft, engines, missiles, ground support equipment, and plant/fleet equipment is performed. In addition, failed component parts from aircraft are repaired.

Although the nature and complexity of the work involved in the different types of overhauls and repairs may be quite diverse with respect to trade-skill and equipment requirements, each type of rework can be described by a unique network of disassembly, repair, and reassembly requirements at the various work centers or shops. A simple example of such a network of routings of a small component part as shown in Figure 1.1. Networks of complex products such as aircraft may consist of thousands of operations.

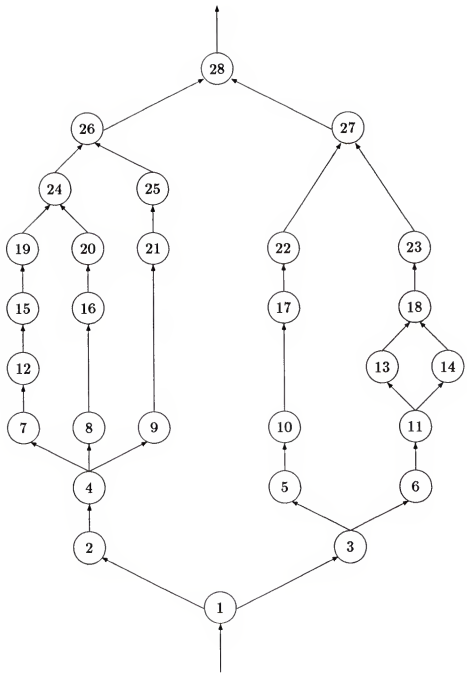


Figure 1.1: Routing Network of a Small Component Part

Since the actual disassembly-reassembly networks provide all the necessary information on the overhaul or repair to be performed, the distinction between aircraft, engines, missiles, etc., need no longer be made. Henceforth an overhaul or repair will simply be referred to as a job.

Jobs may arrive at a NARF in a variety of ways. However, regardless of the nature of the arrival to the system, jobs compete for resources of manpower (trade- skill) and machines at the various shops within a NARF. These shops are generally operated at near or above capacity limits so that backlogs of work-in-process almost always exist.

A NARF is by no means a small operation. A typical NARF will have over 100 production shops, handling thousands of jobs simultaneously, resulting in on the order of tens of thousands of shop operations or service completions daily. The situation is a very large, complex management problem. Finally, the purpose is to provide timely, cost-effective repairs so as to maintain fleet readiness at a maximum. Thus, a very visible measure of a NARF's performance is the ability or inability to meet scheduled completion dates and thereby return repaired items to the fleet on time.

A major concern for NARF managers is the early detection and expeditious handling of any jobs that have fallen behind schedule. Currently available management information systems can provide data on the current status of any job. That is, data are collected that allow a manager to determine what work along the branches of a job's disassembly-reassembly network has been completed and thereby determine if that job is behind schedule or not. However, this type of information only allows managers to try to fight "fires" as they occur; that is, they don't find out that a job is in danger of being late until it is already in trouble. As a result, managers are often hurriedly expediting critical elements in the rework process of jobs that are behind schedule to try to minimize the delay in the job's overall completion time.

A more appropriate tool for these managers would be an information system that could predict the "fires" before they actually occur. Such a system would look forward in time, taking into account the status of all work currently in process, as well as, any arriving jobs during the time frame considered. The system would project the completion time for any job and provide information on any potential bottlenecks, i.e., critically

overloaded shops. In particular, management would obtain an expected completion time for a job of interest, as well as, the probability that it will exceed its original target completion date by a given amount of time.

Given such information, management could then respond to any unsatisfactory completions and bottlenecks by raising job priorities, buying overtime, or shifting manpower, and thereby averting a potential "fire," i.e., prevention as opposed to reaction.

Accuracy of job completion and bottleneck projections, though certainly important, is not the only criterion of such a system. Absolutely accurate results that are not timely and cost effective are practically worthless. A manager needs information today to avoid a potential fire a few days later. Therefore, awkward, complex analytic models or cumbersome, time-consuming detailed simulations are of little use. Instead, relatively fast yet reasonably accurate approximation techniques are required.

The immediate goal of this research, however, is not to solve problems which are as large and complex as those encountered at a NARF. The reasons for this are fairly obvious. In order to evaluate the accuracy of heuristic solution procedures, the true solution (or some very reliable

approximation) must be attainable at a reasonable cost of time and money. As will be made clear in Chapter 2 of this dissertation, the existing theory of queueing networks does not include exact solutions to problems of this nature. Basically this leaves only Monte-Carlo simulation as a reliable tool for an approximation. The cost of the development and execution of a large simulation does not seem justifiable for evaluating an, as yet, untested heuristic.

Instead, a more reasonable approach is to begin with much smaller models which still exhibit many of the salient features of NARF-like facilities and slowly build up size and complexity. Such a method allows the insights gained in solving smaller, simpler models to be applied to the larger, more complex ones.

1.3 Scope

This dissertation presents a study of heuristic methodologies for determining sojourn-time distributions of particular jobs in networks of queues. The types of networks considered include serial and branching (with disassembly and reassembly) where jobs cannot be overtaken. The realm of the

problems studied is transient as opposed to steady state. This is because of the desire to study sojourn times under "given scenario" conditions.

An exact technique based on previous work is developed and evaluated revealing the computational intractability for even very small problems. Given the failure of exact techniques, heuristic procedures are presented which can be used in solving very large, complex queueing problems in a computationally efficient mode. These heuristics are discussed in detail including a presentation of the relative time/cost versus accuracy tradeoff for a host of scenarios.

1.4 Overview

In Chapter 2, a review of the current literature is presented. The review is broken into two sections. The first reviews some of the major works dealing with general queueing networks. The second section is a detailed review of sojourn-time results for queueing networks.

Chapter 3 is a detailed presentation of sojourn-time distributions of particular jobs in serial, single-server networks under "given-scenario" conditions. An exact technique based on research by P.G. Harrison [30] is

presented and shown to be impractical in an application mode. Next, heuristic methodologies are introduced. For computational comparison purposes a detailed Monte-Carlo simulation model is also presented. An experimental design strategy is then structured followed by computational results. Finally, these results are analyzed and conclusions as to the performance of the heuristic are offered.

Simple networks of disassembly, repair, and reassembly type operations are considered in Chapter 4. The results of the analysis of an exact solution procedure for the serial case make it quite obvious that such a procedure in this case is also intractible. Therefore, only heuristic procedures are considered for evaluation. The layout of this section is structured much like Chapter 3 in that both the heuristics and the detailed Monte-Carlo simulation model are presented followed by an experimental design. Finally, results and conclusions are detailed.

The final chapter summarizes the results of the previous sections and provides suggestions for continued research in this area.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This survey is organized into two sections. The first section contains a historical perspective of the field of queueing networks. It is not meant to be an all inclusive review of the field since such a review would necessarily be much too broad for this dissertation. Instead, those works which have had some bearing on sojourn-time results will be covered.

The second section deals with those sojourn-time results which are related to the problem described in Section 1.1. These results reveal a large gap in the literature a portion of which this dissertation addresses.

2.2 Network Queueing Review

As mentioned in the Section 1.1, J.R. Jackson was among the pioneers in the field of queueing networks. His classic research [33] deals with the

development of steady-state distributions of waiting-line lengths in a network of M service centers where center m contains n_m identical servers each working at a rate μ_m with exponentially distributed service times. Customers arrive at service center m in one of two ways: either externally as the realization of a Poisson input stream with rate λ_m or as an internal transfer from another center. Once served at center m a customer instantaneously transfers to center k ($k = 1, 2, ..., M$) with probability Θ_{km} and exits the system with probability $1 - \sum_{k=1}^{M} \Theta_{km}$. The average arrival rate of customers to center m from any source is defined as Γ_m. In steady state, it is easily seen that

$$\Gamma_m = \lambda_m + \sum_{k=1}^{M} \Theta_{mk}\Gamma_k, \qquad m = 1, ..., M.$$

If L_m denotes the number of customers waiting and in service at center m and $L = L_m, m = 1, ..., M$ defines the state of system, then Jackson proves that the steady state distribution of the state of the system is

$$Pr[L] = \prod_{m=1}^{M} P_L^m,$$

as long as

$$\Gamma_m < n_m\mu_m, \qquad m = 1, ..., M,$$

where

$$P_L^m = P_o^m(\Gamma_m/\mu_m)^{L_m}/L_m! \qquad 0 \leq l_m \leq n_m$$

$$P_o^m(\Gamma_m/\mu_m)^{l_m}/n_m! n_m^{l_m - n_m} \qquad l_m \geq n_{m+1}.$$

This means that in steady state the system behaves as if its service centers were independent, i.e., exactly like the simple individual multi-server queues they resemble and thus may be analyzed in isolation. This work provided the basis for analysis of these so-called Jackson networks.

Jackson's results were not surprising given the work of Burke [5] and Reich [56]. Burke [5] proved that the output of an M/M/S queue in equilibrium is Markovian (exponential) with the same rate as the input. He also proved that for tandem Markovian queues the queue length and sojourn-time at the second stage is independent of the first stage. Reich [56] got similar results using a reversibility argument. Burke and Reich's results formalized observations made by R.R.P. Jackson [35] in one of the first papers on tandem queues. He noted an "apparent" independence of queue length distributions in the equilibrium case in deriving state probabilities.

J.R. Jackson extended his own work [34] based on the observation that in real production systems the rate at which new work is injected tends

to decrease and the service rate tends to increase as the amount of work-in-process grows. In this work, Jackson modeled the arrival process as a generalized Poisson arrival process where potential customers are generated by a stable Poisson process but the probability that the customer may enter the system depends upon the number of customers currently there. The service centers are modeled as generalized service-completion processes where the number of servers functioning at a center is based on the queue length. The results of this study are steady-state queue length distributions, which, again demonstrated the independence property in the product form of the solution.

Gordon and Newell [22] considered closed Jackson networks where a fixed number of customers traverse the network, that is, customers neither arrive nor depart the system. Such systems are shown to be stochastically equivalent to an open system where the number of customers cannot exceed N.

Kelly [37] considered open and closed networks with different customer classes and multi-server, infinite server and processor sharing service centers. He established equilibrium queue length distributions for the open and

closed cases. He also used time reversal to obtain additional information on the process considered.

The models discussed so far are all limited to exponential service time distributions. Baskett, Chandy, Muntz, and Palacious (BCMP) [1] attacked this limitation for open, closed, and mixed queueing networks in the context of computer systems modeling. A finite number of customer classes is considered where customers transform from one class to another according to a given transition probability. External arrivals to the system are from Poisson processes. The differences between classes are marked by different routes and demands upon the system. In addition to different customer classes, four types of service centers are considered.

In type 1 centers there is a single server and the service discipline is first-come, first-served. Service time distribution is exponential regardless of customer class and is dependent on queue length.

Type 2 centers consist of a single server which simultaneously provides service to all customers. In other words, each customer receives partial service every time period. Distinct service time distributions with rational Laplace transforms are considered for each customer class.

Type 3 centers exhibit no queueing; i.e., the number of servers is always greater than or equal to the number of customers at that center. As with type 2 centers, distinct service time distributions with rational Laplace transforms are considered.

Finally, type 4 centers are single server systems with preemptive-resume, last-come-first-served service discipline. Again service time distributions with rational Laplace transforms are assumed.

These are the so-called BCMP networks. The results of this research are the development of joint equilibrium queue length distributions. The analysis technique used is based on Whittle's concept of independent balance [70,71] that simplifies the problem of finding steady-state solutions. Chandy [8] also describes this technique and calls it the principal of balance.

Kelly [38] obtained results similar to Baskett et al. [1] for a system having several classes of customers where each class has a route through the network that depends upon the customers past route. He also considered the case where a customer's service time at a queue is dependent upon its route and service requirements at other queues in the route. Kelly showed

that the queue length distribution, while not in general Markovian, has a simple equilibrium form.

Buzen [7] presented computational methods for deriving equilibrium distributions of customers in closed queueing networks with exponential servers. The algorithms presented are based on two-dimensional iterative schemes which are fairly easy to implement and computationally efficient.

Lemoine [46] provided an excellent survey of equilibrium analysis of networks of queues prior to 1977. He covered a wide variey of topics including Whittle's partial balance [70,71], departure processes, and time reversal, as well as, sojourn times for customers. However, he points out that the sojourn-time distribution for an arbitrary customer in an open network is a very difficult problem.

The development of exact analytic solutions of models of more general systems than those described so far has been limited by the additional complexity. Shantikumar and Buzacott [63] point out that all attempts of exact solution of queueing networks of dynamic job shops with service disciplines other than first-come-first-served, preemptive-resume, last-come-first-served, processor sharing and no queueing have failed. Also, service

time disciplines other than exponential and ones with rational Laplace transforms lead to problems that are basically intractible.

The failure of exact analytic models and the time/cost of detailed simulation studies for general queueing networks with more general service time distributions and service disciplines led to the development of several approximation methods. Chandy, Herzog and Woo [9] considered a local balance technique based upon an application of Norton's theorem from electrical circuit theory (see e.g. Smith [65] p. 209). This technique determines approximate queue lengths and waiting time distributions for closed queueing networks with first-come-first-served discipline and general service time distributions. However, the majority of the approximation techniques can be classified as either decomposition or diffusion methods.

The technique of decomposition is based on the results of Jackson [33] described earlier. That is, under certain circumstances, (i.e., Poisson arrivals, service time distributions with rational Laplace transforms, and service disciplines as described by Baskett et al. [1]), service centers can be analyzed separately and exact results can be obtained. Beyond these circumstances decomposition has been used as an approximation method.

This can be done by either considering the related input and output processes of subsystems or by separaton of the total system into a hierarchy of *aggregate systems* with only a few interactions between the various levels. Disney and Cherry [15], Kuehn [43], Shantikumar and Buzacott [63] employed the former while Courtois [12] used the latter. However, it should be noted that these techniques have only been applied for steady-state results.

The second vein of approximation techniques applies diffusion theory and Brownian motion. These techniques are based upon the property that nearly saturated queues are rather insensitive to the detailed form of the arrival or service distributions. This property has been investigated by Kingman [40] and Iglehart and Whitt [32]. A survey of these and other related research is presented by Whitt [69].

These saturated queues or heavy traffic approximation theorems rely on the central limit theorem as do diffusion approximations. Both assume that the number of events in a given time period is normally distributed.

Early work with diffusion approximations can be found in Feller [17]. Newell [53] provided an excellent introduction to the use of diffusion theory in the study of discrete processes. Applications to congestion theory for

single queues have been shown by several researchers. A diffusion model to the waiting time in a M/G/1 queue was applied in Gaver [18] and to the analysis of backlog delays for remote terminals in Gaver [19]. Gaver and Shedler [20] developed a procedure for evaluating CPU utilization in a multiprogramming environment modeled as a cyclic queueing system. J.M. Harrison developed a diffusion-based limit theorem for single queues [27] and applied a diffusion approximation for tandem queues [28]. In this paper, Harrison proved a limit theorem for a network of two single-server queues in tandem justifying a heavy traffic approximation for the sojourn-time distribution. He showed that this approximation is a diffusion process that behaves like two-dimensional Brownian motion with reflecting barriers at the coordinate axes in Harrison [29]. A small example is solved explicitly but Harrison pointed out that significant analytical difficulties must be overcome before a general solution can be found.

Diffusion techniques for other forms of queueing networks have also been applied. J.M. Harrison [25,26] and Wenocur [68] considered single, assembly-like queues as models of production assembly processes.

Reiman [58] developed limit theorems for general queueing networks. Reiser and Kobayashi [59] discussed the accuracy of diffusion approximations of equilibrium queue size distributions for Jackson-like queueing networks of the type described earlier (Jackson [34]) with general arrival and service distributions. The major result of this work was that under heavy traffic situations diffusion proved more accurate than exponential service models.

At about the same time Kobayashi [41,42] presented similar results for equilibrium and transient cases, respectively. In the equilibrium study, the queue lengths for all the nodes are modeled as a vector-valued Wiener process (Brownian motion)with a reflecting barrier to prevent negative lengths. A closed form expression for the covariance matrix is derived using the means and variances of the service distributions along with the transition probability of the network. The transient results include methods for projecting throughput, utilization, and average response times; however, varying customer types and service disciplines other than first-come-first-served were not considered.

This concludes the review of the major research in general network queueing theory which relates to the problems considered in this dissertation. In the next section, existing sojourn-time results are presented.

2.3 Sojourn-Time Results Review

From a customer's viewpoint, a sojourn or passage time is one of the most important aspects of performance of networks of queues. However, until recently, very little has been done to advance the theory in this area. The purpose of this section is to present these analysis techniques and to show the limitations with respect to the problem described in Section 1.1.

Some of the first results on sojourn times in queueing networks were presented by Reich [56,57]. He proved that for a series of M/M/1 queues with first-come-first-served (FCFS) queue discipline, the steady-state sojourn times at each queue are independent. Burke [6] provided an alternative proof based upon a reversibility argument and the observation that customers cannot be overtaken or passed by later arriving customers.

This overtaking condition was shown by Takacs [66] to bring on dependencies among the partial sojourn times incurred at individuals nodes.

These dependencies lead to enormous difficulties in analysis. Simon and Foley [64] and Mitrani [51], also demonstrated this property in correcting assumptions made by Lemoine [46]. Lemoine's work, though not valid for general networks because of possible overtaking, did extend Reich's results for serial queues (Reich [57]) to tree-like Jackson networks with single server queues which do not permit overtaking.

Wong [72] purported to extend Reich's results to single-server, cycleless BCMP networks (Baskett et al. [1]). However, as Melamed [48] pointed out, Wong's results are only true for such networks where overtaking is not permitted in forward short circuits.

P.G. Harrison [30] developed an approximation scheme for sojourn-time distribution for an arbitrary customer under an arbitrary state of the system for a series of M (M/M/1) queues and for tree-like M/M/1 queues. The scheme relys on the steady state queue length distribution and a discrete approximation for the convolution of exponential distributions. In the derivation of the technique, Harrison developed an exact technique for the serial case and he extended this to the equilibrium tree-like case in Harrison [31]. These results will be discussed in greater detail in Chapter 3.

Chow [10] developed the cycle time distribution for a closed network of two FCFS exponential servers. The main result was that if the number of customers, N, in the network is large enough, the cycle time distribution is not sensitive to the ratio of the service rates and it approaches a generalized Erlang as $N \rightarrow \infty$. He also showed that if the service rates are equal the cycle time is exactly Erlang for any N. These results are not surprising given the results of Gordon and Newell [22]. Schassburger and Daduna [60] generalized Chow's results [10] to arbitrarily many nodes using a different method.

Response-time distributions for customers in an M/M/1 queue with feedback (Figure 2.1) are characterized by Lam and Shankar [44]. Multiple classes of customers are considered where the number of rounds of service required is class dependent but otherwise generally distributed.

Walrand and Varaija [67] considered an open multiclass Jackson network in equilibrium. They show that for a customer travelling along a path without overtaking, the sojourn times at the nodes in the path are mutually independent provided service rates are constant along the path.

The sojourn time at a node is shown to have the same distribution as that of a customer in an M/M/1 queue in equilibrium.

They also proved an assertion made by Burke [6] that in a single customer class, 3-node network (Figure 2.2), the sojourn times at the various nodes are not all independent. In other words, a correlation exists between sojourn times at the individual nodes of the network.

In addition, they showed that for any network with a single customer class, the sojourn times along a path with overtaking are not independent.

The overtaking condition was considered by Fayolle, Iasnogorodski, and Mitrani [16]. They find the equilibrium sojourn-time distribution for the network in Figure 2.2. The method is based on the solution of a boundary value problem on a closed contour, which is in turn, reduced to the solution of an integral equation.

Melamed [48] put together a variety of results concerning various aspects of sojourn times for a fairly general class of queueing networks. Included are such things as classes of customers, class changes, class dependent random routing, and state dependent arrival and service rates. Basically, he combined the features of Kelly networks [37,38] and BCMP networks

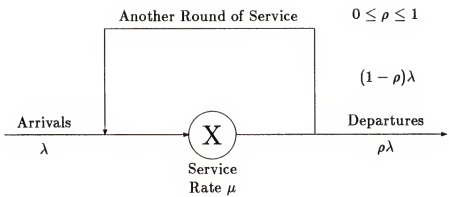


Figure 2.1: M/M/1 Queue with Feedback

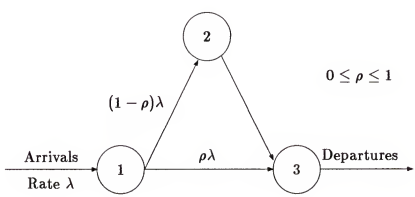


Figure 2.2: Network with Overtaking

(Baskett et al. [1]), and provided equilibrium state distributions, mean sojourn times and sojourn time distributions along overtake free paths.

Daduna [14] showed that for customers in a multi-class closed Gordon-Newell [22] network with single exponential service nodes, the steady-state passage time along an overtake free path is a generalized hyper-Erlang where the mixing distribution is given by the steady-state behavior of the network at arrival times on the path. He extended these results in Daduna [13] to include the case where the first and last nodes can be multiserver queues.

Boxma and Donk [2] determined the joint distribution of consecutive sojourn times of a customer in a closed cycle of two single-server exponential queues. The result is the Laplace-Stieltjes transform of the distribution which is proved using a reversibility argument. These results and the results of Schassburger and Daduna [60] are generalized by Boxma, Kelly, and Konheim [3] for a closed cycle of arbitrarily many single-server queues.

A recent paper by Kelly and Pollett [39] extended and unified the results of Daduna [14] and Boxma et al. [3]. They find the joint distribution of consecutive sojourn times for a customer along an overtake free path in a

closed multiclass Jackson network. In this case the individual sojourn times are not independent but their joint distribution has a relatively simple representation in terms of the product form of the stationary state distribution at an arrival instant.

Packet-based data collection systems were modeled by Shalmon and Kaplan [62] as an open tandem network of capacitated queues with deterministic service and multiple, interfering sources. Sojourn-time distributions for arbitrary customers traversing the network in equilibrium are presented. The solution is the joint moment-generating function for the case where queue length capacities do not increase in the direction of flow.

The technique of randomization has been used recently to obtain sojourn results. The method has been known for some time. Cohen [11], Grassman [23,24], Lippman [47], Serfozo [61], and Keilson [36] described and utilized it in various forms and applications. The technique can be explained by considering a continuous time Markov process with bounded infinitesimal generator matrix Q. The state transition matrix, $\mathbf{P}_t$, is by definition

$$\mathbf{P}_t = e_t^{\mathbf{Q}} = \sum_{n=0}^{\infty} \frac{t^n}{n!} \mathbf{Q}^n, \tag{2.1}$$

or equivalently,

$$\mathbf{P}_t = e^{-ft} \sum_{n=0}^{\infty} \frac{f^n t^n}{n!} \mathbf{R}^n, \text{ where} \tag{2.2}$$

$\mathbf{R} = \mathbf{I} + \mathbf{Q}/f$, and

$f = max_i[q_{ii}]$, $q_{ii} = i^{th}$ diagonal element of Q.

The advantage of Equation 2.2 over Equation 2.1 is that R is a stochastic matrix while Q is not. As a result, the computation of Equation 2.2 is stable whereas Equation 2.1 may not be. Also, R defines a discrete Poisson process. Thus, the holding time of each state is exponential which makes the analysis much easier.

For computational purposes, Equation 2.2 can be approximated by truncating the infinite series to some arbitrary number of terms, N; i.e.,

$$\mathbf{P}_t \approx e^{-ft} \sum_{n=0}^{N} \frac{f^n t^n}{n!} \mathbf{R}^n.$$

This is the foundation of randomization heuristics.

Melamed and Yadin [49] developed a randomization based procedure for computing numerical bounds on cumulative-time distributions over discrete state Markov processes. An application to sojourn-time distributions is presented and results compared for a small problem with known distribution. This technique also handles overtaking conditions.

Melamed and Yadin [50] also investigated the computational aspects of this technique. They developed an optimal storage scheme for open Jackson networks along with improved approximations for bounding curves.

The technique appears to give good results. The sparsity of the transition matrix lends itself to matrix techniques which exploit this structure, as demonstrated by Melamed [49,50]. However, it still falls prey to computation time since the state spare explodes as the networks considered grow.

The bulk of the work reviewed to this point has either been exact techniques or approximations based on analytic results. A noteworthy exception to this is a simulation model developed by Bulkin, Colley, and Steinhoff [4]. Their work is concerned with projecting and sequencing shop operations for the next shift in job shops at the El Segundo Division of Hughes Aircraft Corporation. The objective of the study was to provide shop foremen, in advance, the information required to determine the priority of jobs which should come through his shop during the next work period and the tools, raw material, etc., required to work them.

The model was a success as far as accuracy of results was concerned; however, the time and cost involved in the simulation proved to be a prohibitive factor. This study was done in the early 1960s. The speed of today's computers has certainly increased dramatically while costs have come down; thus, for reasonably sized problems, simulation is certainly a viable alternative. However, for problems the size of the NARF as described in Section 1.2, the same limitations still hold.

It should be fairly clear that with the exception of the work by P.G. Harrison [30] and the simulation by Bulkin et al. [4] sojourn-time results for arbitrary customers under "given scenario" conditions have been largely ignored. This is quite understandable given the severe difficulty of obtaining analytic results in other than very simple cases. Research on networks of queues where jobs may be disassembled and later reassembled is apparently lacking altogether, for the same reason as above.

The contribution of this research is the development of simple heuristics to solve these problems which are accurate and yet computationally attractive for application to real large-scale problems.

As will be illustrated in the remaining chapters the contributions of this research is the development of simple heuristics to solve these problems. The heuristics are shown to be accurate while retaining the important property of computational efficiency which allows application to real large-scale problems.

CHAPTER 3

SERIAL QUEUEING MODELS

3.1 Introduction

A series network of queues is probably the simplest form of a queueing network. Yet, as pointed out in Chapter 2, the problem of specifying sojourn-time distributions for such networks has only been solved for a very limited range of arrival and service time distributions and, generally speaking, under equilibrium or steady-state conditions where jobs cannot be overtaken.

Often for those cases where exact solutions do exist, application to even moderately sized "real-world" problems can be an impractical if not infeasible task. Many of the solutions are available only in transform domains where reinversion to the time domain is unattainable in generalized form.

However, it is often the case that the actual distribution of the desired performance measure is not explicitly required. Instead, the mean,

variance, and possibly tail probabilities are of practical importance. The mean and variance are at least theoretically attainable directly from the transform of the desired distribution through differentiation. In practice, the size and structure of the transform directly relate to the computation effort required to attain these values.

A recent heuristic presented by Platzman, Ammons and Bartholdi [54] provides a relatively simple means for evaluating the probability that a random variable exceeds a specified value, given only its transform. This technique, discussed in detail later, may help overcome the problem of determining tail probabilities. However, computational problems still persist for cases where the transform contains a large number of terms.

In this section, exact and heuristic methodologies for determining sojourn distributions for particular customers in a serial network of single server queues are investigated. Particular interest is paid to the case where the service rates at the various stations are nearly equal and the initial service queue lengths are significant (i.e., a large number of customers in a relatively balanced system).

This case is emphasized because of the desire to study large job shop scenarios like the NARF described in Chapter 1. These systems are characterized by heavy customer (job) traffic and non-empty service queues. This situation exists because of management's desire to keep work on hand to minimize server idle time.

The formal definition of the model of such a system is described in Section 3.2. In this section, an exact transform solution is also presented. However, the number of terms in the transform is shown to grow at such a rate that even relatively small problems quickly become intractable.

Given the computational difficulties of the exact methodology, relatively simple heuristic procedures are presented in Section 3.3. A detailed Monte-Carlo simulation model is also developed to provide a basis for assessing the viability of the heuristics in both an accuracy and CPU resource utilization mode. An experimental design for evaluating the heuristics is provided in Section 3.4. In Section 3.5, the results and subsequent analysis of the experimentation are presented. Finally, conclusions are offered in Section 3.6.

3.2 Exact Analytic Solution

In this section, an exact analytic technique for specifying the sojourn-time distribution of particular jobs in a serial network of queues is described. The technique is a generalization of a method presented by P.G. Harrison [30], in that Harrison looked only at exponential service distributions, whereas, phase-type exponential distributions are considered here.

3.2.1 Problem Specification

The problem can be described as a series configuration of M service centers (see Figure 3.1), each having an FCFS (First-Come-First-Served), single server queue. The service times at center j, $(j = 1, ..., M)$ form a sequence of i.i.d. random variables of the generalized Erlang type having p_j phases. The rate of the k^{th} stage is constant and equal to μ_{jk}.

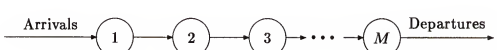


Figure 3.1: Serial Network of Single-Server Queues

The following definitions are necessary.

Let

$n_j(t)$ = number of jobs in the queue and in service at center j at time t.

$N(t)$ = vector representation of all $n_j(t)$
= $[n_1(t), n_2(t), ..., n_M(t)]$.

$r_j(t)$ = phase of service in progress at center j at time t.
Define $r_j(t) = 0$ if server k is idle at time t.
NOTE: $0 \leq r_j(t) \leq p_j$.

$R(t)$ = vector representation of all $r_j(t)$
= $[r_1(t), r_2(t), ..., r_M(t)]$.

$S(t)$ = state of the system at time t
= $[N(t), R(t)]$.

It should be noted that the state of the system is completely specified by $N(t)$ and $R(t)$. Due to the memoryless property of the exponential distribution it is not necessary to know the time that the current phase of service began at each server. Instead only the phase itself needs to be known.

Therefore, since service times in the model described are phased exponentials, all that need be known is the phase of service in progress in order to specify the distribution of remaining service time for a job in service.

3.2.2 Solution Procedure

Given the problem as described above, the requirement is to specify the sojourn-time distribution (or remaining sojourn-time distribution) for a particular arriving job (or for any job currently in the system) which finds the system in state $S(t_0)$. Notice that any jobs behind the particular job will not affect its sojourn-time since the service rates are constant and service discipline is FCFS. Therefore, those jobs at service centers where the particular job has already received service can be eliminated from further consideration. Any jobs queued behind the particular job may also be eliminated. Notice that the particular job need not be marked since it is easily identified as the last job in the system.

Let $\Omega(S)$ define the state space of the system. Explicitly,

$$\begin{aligned}\Omega(S) &= \{N(t) \mid 0 \le n_j(t) \le n_j(t_0) + \textstyle\sum_{1\le k\le j-1}(n_k(t_0) - n_k(t)),\\ &\qquad R(t) \mid 0 \le r_j(t) \le p_j, j = 1, ..., M\}.\end{aligned}$$

Note that $\mid \Omega(S) \mid$ is finite, where $\mid \mathbf{A} \mid$ = cardinality of set $\mathbf{A}$.

Now let $\mathbf{T}$ denote the transition probability matrix where

$T_{\mathbf{uv}}$ = Pr[Next state is $\mathbf{v}$ | Present state is $\mathbf{u}$, $\mathbf{u}, \mathbf{v} \in \Omega(S)$].

Given that there are M centers, there is a maximum of M events which could cause a transition out of any state $\mathbf{u}$; that is, a completion of a phase of service at any of the M centers. Thus, the time to transition out of state $\mathbf{u}$ is equal to the minimum of up to M exponential time intervals. It is shown in Appendix A.1 that this time is exponentially distributed as follows:

$$f_{\mathbf{u}}(t) = \lambda_u e^{\lambda_u t}$$

where

$$\lambda_u = \sum_{1 \le j \le M; n_j(t) > 0} \mu_{j r_j^{\mathbf{u}}},$$

and $r_j^{\mathbf{u}}$ = the phase of service in progress at shop j in state $\mathbf{u}$.

Thus, $\lambda_{\mathbf{u}}$ is the holding rate of state $\mathbf{u}$.

Let $f_{\mathbf{u}}^*(s)$ be the Laplace-Stieltjes transform of holding time of state $\mathbf{u}$, $f_{\mathbf{u}}(s)$, i.e.,

$$L[f_{\mathbf{u}}(t)] = \int_0^\infty e^{-st} df_{\mathbf{u}}(t) = \frac{\lambda_{\mathbf{u}}}{s + \lambda_{\mathbf{u}}}$$

Define the functions $\phi(\mathbf{u}, \mathbf{v})$ and $\psi(\mathbf{u}, \mathbf{v})$ such that

$\phi(\mathbf{u}, \mathbf{v})$ = number of the center from which a completion of a phase

of service causes the transition from $\mathbf{u}$ to $\mathbf{v}$ $(\mathbf{u}, \mathbf{v} \in \Omega(S))$ which is 0 if a one-step transition $\mathbf{u} \rightarrow \mathbf{v}$ is not possible.

$\psi(\mathbf{u}, \mathbf{v})$ = number of the phase of service at center $\phi(\mathbf{u}, \mathbf{v})$ which causes a transition from $\mathbf{u}$ to $\mathbf{v}$. $\psi(\mathbf{u}, \mathbf{v})$ is 0 if a one-step transition $\mathbf{u} \rightarrow \mathbf{v}$ is not possible.

Then the transition rate from state $\mathbf{u}$ to $\mathbf{v}$ of the underlying Markov process is

$$T_{\mathbf{uv}} = \frac{\lambda_{\phi(\mathbf{u},\mathbf{v}),\psi(\mathbf{u},\mathbf{v})}}{\lambda_{\mathbf{u}}}$$

where $\mu_{0,0} \equiv 0$. Notice that $T_{\mathbf{uv}}$ is also the probability that a transition out of state $\mathbf{u}$ goes to state $\mathbf{v}$.

For any row in $\mathbf{T}$ the number of non-zero entries will is at most M. That is, the number of events that cause a transition out of a given state is equal to the number of service centers that are active in that state. This is obviously bounded by the total number of service centers, M.

Let $\mathbf{T}^*$ be a modified transition matrix with the following specification.

$$
\begin{aligned}
T^*_{\mathbf{uv}} &= T_{\mathbf{uv}} f^*_{\mathbf{u}}(s) \\
&= \text{weighted Laplace transform of the holding time} \\
&\quad \text{distribution of state } \mathbf{u} \text{ given the next state is } \mathbf{v}, \\
&= \frac{\mu_{\phi(\mathbf{u},\mathbf{v}),\psi(\mathbf{u},\mathbf{v})}}{\lambda_{\mathbf{u}}} * \frac{\lambda_{\mathbf{u}}}{s+\lambda_{\mathbf{u}}} \\
&= \frac{\mu_{\phi(\mathbf{u},\mathbf{v}),\psi(\mathbf{u},\mathbf{v})}}{s+\lambda_{\mathbf{u}}}.
\end{aligned}
$$

Now, let the cumulative distribution function for the time for the system to transition from state α to state $\beta(\alpha, \beta \in \Omega(S))$ be $G_{\alpha\beta}(t)$ and let $G^*_{\alpha\beta}(s)$ be its Laplace- Stieltjes transform. Thus,

$$G^*_{\alpha\beta}(s) = \int_0^\infty e^{-st} dG_{\alpha\beta}(t).$$

Further define $H_{\alpha\beta}$, such that

$$
\begin{aligned}
H_{\alpha\beta} &= \{\mathbf{h}\} \\
&= \{\mathbf{h} = (h_1, h_2, ..., h_n) \mid n \in Z^+; h_1 = \alpha; h_n = \beta; \\
&\quad h_k \in \Omega(S); T_{k,k+1} \neq 0; 1 \leq k < n\} \\
&\quad \text{where } Z^+ \text{ is the set of positive integers.}
\end{aligned}
$$

The set of all possible sequences of state transitions, or paths, from state α to β is represented by $H_{\alpha\beta}$. Define $\mid \mathbf{h} \mid$ as the cardinality of the set of links in the path $\mathbf{h}$ which is equal to the number of state transitions in the

path $\mathbf{h}$ from state α to state β. For a given α, β pair, $\mid \mathbf{h} \mid$ for all $\mathbf{h}$ are equal, i.e., all paths in $H_{\alpha\beta}$ are the same length.

Harrison [30] proves that for the exponential service case,

$$G^*_{\alpha\beta}(s) = (I - \mathbf{T}^*)^{-1}_{\alpha\beta}. \tag{3.2}$$

The proof involves weighting the contributions of every path in H. The proof for the Erlang service case can be done in basically the same way since the state of the system as defined involves states whose holding times are exponential.

However, a different approach is explicitly provided below. By definition

$$G^*_{\alpha\beta}(s) = \sum_{\gamma \in \{\phi(\alpha,\gamma) \neq 0\}} T^*_{\alpha\gamma} G^*_{\gamma\beta}(s) \tag{3.3}$$

i.e., the weighted Laplace transform of the time to transition from α to β is equal to the sum of the products of the weighted Laplace transform of the time for transition from γ to β times the modified transition rate from α to γ where all γ are one-step reachable states from α.

The relationship holds true for any α, β. However, if β is a one-step reachable state from α, then a potential problem arises. In this case,

$$G^*_{\alpha\beta}(s) = T^*_{\alpha\beta} G^*_{\beta\beta}(s).$$

Since by definition

$$G^*_{\alpha\beta}(s) = T^*_{\alpha\beta}$$

for all one-step reachable states β, then $G^*_{\beta\beta}(s) = 1$ must be true for all states β.

However, consider an α, β pair where β is equal to α, then

$$G^*_{\alpha\alpha}(s) = T^*_{\alpha\alpha} G^*_{\alpha\alpha}(s), \tag{3.4}$$

where $G^*_{\alpha\alpha}(s) = 1$ from above, and $T^*_{\alpha\alpha} = 0$, by definition. Substituting these into (3.4) yields

$1 = 0$ which is a contradiction.

Notice that for α, β not equal and not one-step reachable states that (3.3) is equivalent to the dot product of two vectors, a particular row α of $\mathbf{T}^*$ times column β of $G^*(s)$.

Therefore,

$$G^*(s) = \mathbf{T}^* G^*(s) + I \tag{3.5}$$

where the identity matrix I is added to account for the contradiction described above along the diagonal elements.

Manipulating Equation 3.5 yields

$$G^*(s) = (I - \mathbf{T}^*)^{-1} \tag{3.6}$$

and therefore,

$$G^*_{\alpha\beta}(s) = (I - \mathbf{T}^*)^{-1}_{\alpha\beta} \tag{3.7}$$

for any states α, β.

For the system as specified, it is important to note that all states in Ω are transient with the exception of the state corresponding to an empty system (which is a capturing state). This is because arrivals to the model are ignored since they can have no effect on the sojourn-time distribution of the particular job.

Therefore, with an appropriate ordering of the states, it is easily seen that the transition matrix $\mathbf{T}$, and thus $\mathbf{T}^*$ and $(I - \mathbf{T}^*)$ is upper-triangular. Thus $(I - \mathbf{T}^*)^{-1}$ can be solved by simple back substitution.

Each path in $H_{\alpha\beta}$ corresponds to a unique realization of state transitions where each transition interval is exponentially distributed with known rate. Consider one of these paths, $\mathbf{h}$, and let the random variables $X_i (i = 1, \ldots \mid \mathbf{h} \mid)$ represent these exponentially distributed time intervals. Let the random variable Y represent the time to traverse this path. Then

$$Y = \sum_{1 \le i \le |\mathbf{h}|} X_i.$$

Thus, Y is simply a finite sum of exponentially distributed random variables with possibly different rates. This is the well known generalized Erlang distribution.

Therefore, the distribution of $G_{\alpha\beta}(t)$ is the so-called generalized hyper-Erlang, that is, a weighted combination of general Erlang distributions. Each of these general Erlang distributions corresponds to a unique *path* from state α to β. Thus, there is a one-to-one correspondence between the cardinality of H and general Erlang terms in $G_{\alpha\beta}(t)$. The weight associated with each term is the probability that that corresponding path is taken in transitioning from α to β.

3.2.3 A Small Example

Consider a small network of the type shown in Figure 3.1. Let $M = 2$ and let the service distribution of servers 1 and 2 be degenerate Erlang, i.e., exponential, with rates μ_1 and μ_2, respectively. Thus,

$$p_j = 1 \text{ for } j = 1, 2.$$

Since server j is either idle (if $n_j(t) = 0$) or busy (if $n_j(t) > 0$) and since

$p_j = 1$, $R(t)$ is not required to specify the state of system. This is because $r_j(t)$ will either be 0 or 1 depending upon whether $n_j(t) = 0$ or $n_j(t) > 0$ where $n_j(t)$ is the number of jobs at center j at time t.

Explicitly,

$$r_j(t) = \begin{cases} 0 & \text{if } n_j(t) = 0, \\ 1 & \text{if } n_j(t) > 0. \end{cases}$$

Therefore, the state of the system $S(t)$ is completely specified by $N(t)$.

Now, for this example let the initial state of the system, $S(0)$, be

$$S(0) = N(0) = (n_1(0), n_2(0)) = (2, 1),$$

Arbitrarily label the jobs: $1, 2, 3$ where job 1 is in service at shop 2, job 2 is in service at shop 1 and job 3 is in the queue at shop 1.

The state space, $\Omega(S)$, for this problem can be specified explicitly as:

$$\Omega(S) = \{(2,1), (2,0), (1,2), (1,1), (1,0), (0,3), (0,2), (0,1), (0,0)\}.$$

The state transition diagram is shown in Figure 3.2.

The nodes represent potential states of the system and the arcs represent potential one-step transitions. For instance, from the state $(2, 1)$ a service

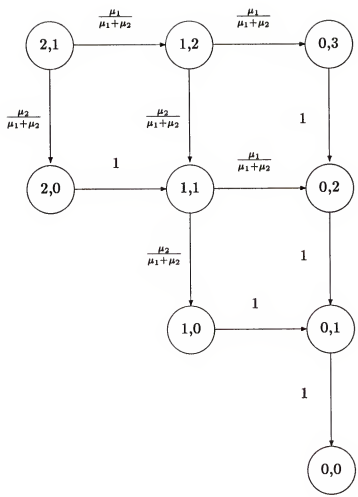


Figure 3.2: State Transition Diagram

NOTE: Arc values represent traversal probabilities

completion by server 1 causes a transition to the state $(1,2)$. Likewise a server 2 completion causes a transition to the state $(2,0)$.

Define W_q as a mapping of the state definition, S, onto the set of counting numbers, where

$$W_q(n_1, n_2) = qn_1 + n_2$$

and

$(n_1, n_2) \in \Omega(S)$, and $q > \sum_{1 \leq i \leq M} n_i(0)$, integer.

Note that q is an arbitrary integer greater than the total number of jobs in the system at time 0. This ensures that the mapping, W_q, is unique for all $(n_1, n_2) \in \Omega(S)$. Table 3.1 below lists the states of the system with the corresponding W_q value and a state label which is arbitrarily assigned sequentially based on decreasing W_q values. (For this example $q = 10$ is used).

The set of paths for the transition from the initial state of the system (state 1) to an empty system (state 9), H_{19} is given below.

$$H_{19} = \{h_k, k = 1, 5\}$$

where

Table 3.1: Example Problem State Labels

State	W_q	Label
(2,1)	21	1
(2,0)	20	2
(1,2)	12	3
(1,1)	11	4
(1,0)	10	5
(0,3)	3	6
(0,2)	2	7
(0,1)	1	8
(0,0)	0	9

$h_1 = (1,3,6,7,8,9)$
$h_2 = (1,3,4,7,8,9)$
$h_3 = (1,3,4,5,8,9)$ Note $| H_{19} | = 5$.
$h_4 = (1,2,4,7,8,9)$
$h_5 = (1,2,4,5,8,9)$

The one-step transition matrix, $\mathbf{T}$, is specified in Table 3.2 and the modified transition matrix, $\mathbf{T}^*$, is given in Table 3.3.

Now, given the above specifications, the problem is to determine the sojourn-time distribution for job 3 (the job in the queue at center 1). Explicitly, $G_{1,9}(t)$ is of interest, i.e., the distribution of time for the system to transition from state 1 to state 9. The Laplace-Stieltjes transform of the solution can be obtained via Equation 3.2, i.e.,

$$G^*_{1,9}(s) = (I - \mathbf{T}^*)^{-1}_{1,9}.$$

Notice that only element $(1,9)$ of $(I - \mathbf{T}^*)^{-1}$ is required; however, by standard backward substitution all values in column 9 must be computed in order to obtain this element. The equations to derive these values follow:

$$G^*_{9,9}(s) = 1$$

$$G^*_{8,9}(s) = G^*_{9,9}(s)\frac{\mu_2}{(s+\mu_2)} = \frac{\mu_2}{(s+\mu_2)}$$

$$G^*_{7,9}(s) = G^*_{8,9}(s)\frac{\mu_2}{(s+\mu_2)} = \left(\frac{\mu_2}{s+\mu_2}\right)^2$$

$$G^*_{6,9}(s) = G^*_{7,9}(s)\frac{\mu_2}{(s+\mu_2)} = \left(\frac{\mu_2}{s+\mu_2}\right)^3$$

$$G^*_{5,9}(s) = G^*_{8,9}(s)\frac{\mu_1}{(s+\mu_1)} = \frac{\mu_1\mu_2}{(s+\mu_1)(s+\mu_2)}$$

$$G^*_{4,9}(s) = G^*_{7,9}(s)\frac{\mu_1}{(s+\mu_1+\mu_2)} + G^*_{5,9}(s)\frac{\mu_2}{(s+\mu_1+\mu_2)}$$

$$= \frac{\mu_1\mu_2^2}{(s+\mu_1+\mu_2)(s+\mu_2)^2} + \frac{\mu_1\mu_2^2}{(s+\mu_1+\mu_2)(s+\mu_1)(s+\mu_2)}$$

$$G^*_{3,9}(s) = G^*_{6,9}(s)\frac{\mu_1}{(s+\mu_1+\mu_2)} + G^*_{4,9}(s)\frac{\mu_2}{(s+\mu_1+\mu_2)}$$

$$= \frac{\mu_1\mu_2^3}{(s+\mu_1+\mu_2)(s+\mu_2)^3} + \frac{\mu_1\mu_2^3}{(s+\mu_1+\mu_2)^2(s+\mu_2)^2} + \frac{\mu_1\mu_2^3}{(s+\mu_1+\mu_2)^2(s+\mu_1)(s+\mu_2)}$$

$$G^*_{2,9}(s) = G^*_{4,9}(s)\frac{\mu_1}{(s+\mu_1)}$$

$$= \frac{\mu_1^2\mu_2^2}{(s+\mu_1+\mu_2)(s+\mu_1)(s+\mu_2)^2} + \frac{\mu_1^2\mu_2^2}{(s+\mu_1+\mu_2)(s+\mu_1)^2(s+\mu_2)}$$

$$G^*_{1,9}(s) = G^*_{3,9}(s)\frac{\mu_1}{(s+\mu_1+\mu_2)} + G^*_{2,9}(s)\frac{\mu_2}{(s+\mu_1+\mu_2)}$$

$$= \frac{\mu_1^2\mu_2^3}{(s+\mu_1+\mu_2)^2(s+\mu_2)^3} + \frac{\mu_1^2\mu_2^3}{(s+\mu_1+\mu_2)^3(s+\mu_2)^2} +$$

$$\frac{\mu_1^2\mu_2^3}{(s+\mu_1+\mu_2)^3(s+\mu_1)(s+\mu_2)} + \frac{\mu_1^2\mu_2^3}{(s+\mu_1+\mu_2)^2(s+\mu_1)(s+\mu_2)^2} +$$

$$\frac{\mu_1^2\mu_2^3}{(s+\mu_1+\mu_2)^2(s+\mu_1)^2(s+\mu_2)}$$

$$= \mu_1^2\mu_2^3\Big(\frac{1}{(s+\mu_1+\mu_2)^2(s+\mu_2)^3} + \frac{1}{(s+\mu_1+\mu_2)^2(s+\mu_1)(s+\mu_2)^2} + \frac{1}{(s+\mu_1+\mu_2)^2(s+\mu_1)^2(s+\mu_2)} +$$

$$\frac{1}{(s+\mu_1+\mu_2)^3(s+\mu_2)^2} + \frac{1}{(s+\mu_1+\mu_2)^3(s+\mu_1)(s+\mu_2)^2)}$$

Table 3.2: One-step Transition Matrix, **T**

	To State								
From State	1	2	3	4	5	6	7	8	9
1	-	$\frac{\mu_2}{\overline{\mu_1+\mu_2}}$	$\frac{\mu_1}{\overline{\mu_1+\mu_2}}$	-	-	-	-	-	-
2	-	-	-	$\frac{\mu_1}{\overline{\mu_1}}$	-	-	-	-	-
3	-	-	-	$\frac{\mu_2}{\overline{\mu_1+\mu_2}}$	-	$\frac{\mu_1}{\overline{\mu_1+\mu_2}}$	-	-	-
4	-	-	-	-	$\frac{\mu_2}{\overline{\mu_1+\mu_2}}$	-	$\frac{\mu_1}{\overline{\mu_1+\mu_2}}$	-	-
5	-	-	-	-	-	-	-	$\frac{\mu_1}{\overline{\mu_1}}$	-
6	-	-	-	-	-	-	$\frac{\mu_2}{\overline{\mu_2}}$	-	-
7	-	-	-	-	-	-	-	$\frac{\mu_2}{\overline{\mu_2}}$	-
8	-	-	-	-	-	-	-	-	$\frac{\mu_2}{\overline{\mu_2}}$
9	-	-	-	-	-	-	-	-	-

Table 3.3: Modified State Transition Matrix, $\mathbf{T}^*$

	To State								
From State	1	2	3	4	5	6	7	8	9
1	-	$\frac{\mu_2}{s+\lambda}$	$\frac{\mu_1}{s+\lambda}$	-	-	-	-	-	-
2	-	-	-	$\frac{\mu_1}{s+\mu_1}$	-	-	-	-	-
3	-	-	-	$\frac{\mu_2}{s+\lambda}$	-	$\frac{\mu_1}{s+\lambda}$	-	-	-
4	-	-	-	-	$\frac{\mu_2}{s+\lambda}$	-	$\frac{\mu_1}{s+\lambda}$	-	-
5	-	-	-	-	-	-	-	$\frac{\mu_1}{s+\mu_1}$	-
6	-	-	-	-	-	-	$\frac{\mu_2}{s+\mu_2}$	-	-
7	-	-	-	-	-	-	-	$\frac{\mu_2}{s+\mu_2}$	-
8	-	-	-	-	-	-	-	-	$\frac{\mu_2}{s+\mu_2}$
9	-	-	-	-	-	-	-	-	-

Note: $\lambda = \mu_1 + \mu_2$

Now, once $G^*_{1,9}(s)$ has been obtained the distribution $G_{1,9}(t)$ can be derived by inverting the transform. Each term in the transform can be manipulated separately using partial fraction expansion to yield simple terms of the form:

$$\sum_{i=1}^{k} A_i \left(\frac{1}{s-a}\right)^i$$

where

A_i = a constant,

a = a state transition rate, and

k = the power of the factor $(s-a)$ in each term.

These terms can be relatively easily inverted back to the time domain yielding exponential terms (see e.g. Muth [52]). This inversion, while conceptually simple, leads to computational problems as the number of terms in $G^*(s)$ grows. For this example these steps are performed in Appendix A.2.

As can be seen from this very simple example, obtaining $G(t)$ can be a difficult task. However, in application, the explicit distribution, $G(t)$, is often not required. Instead, as mentioned previously, the mean, variance, and possibly, the probability that the sojourn time exceeds some threshold values are more appropriate. In this case, reinverting $G^*(s)$ is not

necessary as these values can be obtained directly from the transform. For this purpose, let

$$m_k = \begin{cases} \text{kth moment of } G(t), (G^*(s)) \\ \int_0^\infty t^k G(t)dt \\ (-1)^k[\frac{D^k}{ds^k}G^*(s)]_{s=0} \text{(see e.g., Muth [52], p. 193.)} \end{cases}$$

Thus, μ_s and σ_s^2, the mean and variance of the sojourn distribution are

$$\mu_s = m_1$$

$$\sigma_s^2 = m_2 - m_1^2.$$

The probability, τ, that a random variable, X, exceeds a given value, A, can be represented notationally as

$$\tau = P[X > A] = 1 - F_X(A),$$

where

F_X is the cumulative distribution of X.

Platzman, Ammons, and Bartholdi [54] derived a heuristic for computing $P[X > A]$ when only $F_X^*(s)$, (Laplace-Stieltjes transform of $F_X(x)$), is known. The approximation for τ is

$$\hat{\tau} = \frac{u - A}{u - L} + \sum_{1 \le n \le N} \frac{\alpha^{n^2}}{n\Pi} \Im[(\beta^n - \gamma^n) F_x^*(-n\omega j)] \tag{3.8}$$

where $\Im[a]$ denotes the imaginary part of the complex number, a , and

$\alpha = \exp[-D^2\omega^2/2]$,

$\beta = \exp[-A\omega j] = \cos(A\omega) - j\sin(A\omega)$,

$\gamma = \exp[-U\omega j] = \cos(U\omega) - j\sin(U\omega)$,

$\omega = 2\Pi/(U - L)$,

$j = \sqrt{-1}$,

$L =$ lower bound on $F_X(x)$,

$U =$ upper bound on $F_X(x)$,

$D =$ the accuracy to which X and A are specified, and

$N =$ number of terms in the approximation.

The parameters L, U, D, and N can be chosen in such a way to make $\hat{\tau}$ as close as desired to the true value τ.

The computation of $\hat{\tau}$ can be easily coded requiring only seven memory locations. It should be noted that the difficulty of computing is dependent upon the particular function $F_X^*(s)$. This will be discussed in the next section.

3.2.4 Computational Considerations

While the back substitutional solution of $(I-T^*)^{-1}$ which yields $G^*_{\alpha\beta}(s)$ is a straightforward and simplistic procedure, there still are computational problems to overcome.

The number of back substitutions that must be made to solve $(I-T^*)^{-1}$ is equal to the number of states in $\Omega(S)$. However, the number of terms in the solution of each of these states is equal to the number of paths from each state to the empty state. Thus, the computational effort and storage required depends upon the number of paths from α to β, i.e., $\mid H_{\alpha\beta} \mid$. This value is a function of the number of queues, the number of service phases at each shop, the initial state α, and the final state β.

For simplicity, consider the exponential service case, i.e., $p_j = 1$, for all j. Let $f_{H_{\alpha\beta}}(M) = \mid H_{\alpha\beta} \mid$. This value is sum of the $f_{H_{\gamma\beta}}(M)$ values for all one-step reachable states, γ, from state α. Thus, a recursive relationship exists, which is, explicitly,

$$f_{H_{\alpha\beta}}(M) = \sum_{\gamma \in J_\alpha} f_{H_{\gamma\beta}}(M)$$

where

J_α = set of one-step reachable states from α

$$= \{\gamma \mid \phi(\alpha, \gamma) > 0\}.$$

Recall that $\mid J_\alpha \mid \leq M$.

Consider a simple 2-queue, exponential service system as in the example in the previous section. For such a system notice, from Figure 3.2, that from a given node (state of the system) there are at most 2 one-step reachable nodes. Thus, the number of paths from that node to the empty system node (Node $(0,0)$) is simply the sum of the number paths from each of the one-step reachable nodes to the empty system node. For example, the number of paths from node $(5,3)$ is the number of paths from node $(5,2)$ plus the number of paths from node $(4,4)$.

The number of paths from some state α to another β depends upon the shape of the state space. Table 3.4 below gives representative values for some simple networks. From this table it is easy to see that the number of paths explodes as the queue sizes increase and the number of queues grows.

This explosion leads to storage problems and computational intractability for even relatively small problems.

Table 3.4: Number of Paths in Some Representative Scenarios

Queue Size at Shop:					
1	2	3	4	Number of States	Number of Paths
0	0	5	5	51	1638
0	0	10	5	121	1,225,785
0	0	10	10	176	15,737,865
0	3	3	3	140	678,912
0	5	5	5	506	6.726×10^{10}
0	5	10	5	1001	3.244×10^{14}
0	10	5	5	1386	6.055×10^{17}
0	10	10	10	3311	4.541×10^{23}
2	2	2	2	273	55,099,278
3	3	3	3	969	8.259×10^{12}
4	4	4	4	2530	1.656×10^{18}
5	5	5	5	5481	3.866×10^{24}
10	10	10	10	27636	1.064×10^{38}

All values in this table reflect exponential service at each shop.

3.3 Heuristic Solution Procedures

Motivated by the failure of the exact technique to yield efficient solutions for the serial queueing model, heuristics are developed in this section. First, a Monte-Carlo simulation model is developed for evaluation of its own merits, as well as, for a means of obtaining a "true" sojourn-time distribution for problems where exact solutions are not attainable at a reasonable cost. Following that, two simple heuristics are presented which are founded on the principles of the Central Limit Theorem.

3.3.1 Monte-Carlo Simulation

A simulation model of the serial queueing problem has been developed in FORTRAN. The model has been structured in modular form to facilitate the verification and validation processes.

The model itself is a terminating simulation which is repeatedly run using different random number streams in order to empirically build sojourn-time distributions. The number of repetitions can be gauged to ensure a specified accuracy in matching the true sojourn-time distribution.

The development of a simulation model of a serial shop queueing network is of little academic interest and, therefore, will not be discussed in any detail. The model was validated against the exact distribution through comparison of the mean, variance, and distribution. The specifics of the validation are provided in the results section (Section 3.5.2).

3.3.2 Deterministic Simulation Heuristic (DSH)

The inherent difficulty of the exact analytic solution lies in the vast number of paths that the system may traverse as the particular job is serviced. This, coupled with the correlation or dependence between the individual shop sojourn times (discussed in Chapter 2) make such a solution impractical for all but very small problems.

Perhaps the simplest heuristic scheme is to assume that all service times are deterministic (and equal to the mean of the appropriate service distribution) and make a single pass simulation using these fixed service times. This would give an approximation for the mean sojourn time. The variance could be approximated by simply summing variances. This is the basic scheme of the Deterministic Simulation Heuristic (DSH).

The key difference between this heuristic and the exact solution is that the exact technique makes use of all possible paths that the system may transition through while the heuristic is concerned only with the most probable. The exact solution consists of a weighted combination of all paths and thus the exact sojourn distribution is generalized hyper-Erlang, while for the heuristic it is simply a generalized Erlang.

Motivation for the heuristic. The motivation for DSH is the Central Limit Theorem. The application of this theorem can best be explained by way of a small example.

Consider a 2 queue serial network of exponential servers with nearly equal rates. Assume the particular job is at the back of the first queue and both queues contain several jobs.

Now, the distribution of time that the particular job spends at the first queue is simply the sum of exponentially distributed intervals since the job must wait for all jobs ahead of it to complete service before it can be processed. Thus, the distribution of time at the first shop is an N stage special Erlang where N is the number of jobs at Shop 1.

The distribution of time at the second shop is similar except that there is a probability distribution associated with N; that is, N could take on values from 1 to K, where K is the total number of jobs in the system at the start.

Since the service rates are nearly equal and the number of jobs at each shop is significant (greater than 3), then, by the Central Limit Theorem, the distribution of jobs at the second shop will be relatively symmetric about the mean. In turn, the sojourn-time distribution at the second shop will also be effectively symmetric. Therefore, the heuristic should give good results. In fact, as demonstrated in Section 3.5 and Section 3.6, such a heuristic is robust in that it gives good results for many cases where the service rates are not similar and where the number of jobs is small.

Generalization of the technique to cases of more shops in series is straightforward. The same is true for service distributions of the phased exponential form (generalized Erlang). In fact, as the variance of the service distribution gets tighter the heuristic should give better results since the tail probabilities of the associated distributions of the number of jobs at each shop will get smaller.

Solution algorithm. Implementation of the DSH technique is extremely simple with computational effort of order M where M is the number of shops. The algorithm used for the purposes of this research is presented below.

Definitions: The following definitions are necessary.

$NJOB(j)$ - The maximum number of jobs that could be at shop j upon arrival of the particular job. (Includes the particular job.)

$STAGE(j)$ - The shop sojourn-time estimate for the particular job at shop j.

$SVAR(j)$ - The contribution to the variance of the total sojourn time from the shop sojourn time at shop j.

$NUMC(j)$ - The number of service completions at shop j during the time the particular job is at that shop. This may have a fractional part if a service is in progress upon arrival of the particular job. This fractional part represents the remaining average service time divided by the average service time.

p_j - the number of service phases at shop j.

μ_{ij} - the service rate of the ith phase of service at shop j.

$\hat{\mu_j}$ - total service rate at shop j, i.e.,

$$\hat{\mu}_j = \frac{1}{\sum_{1 \le i \le p_j} \frac{1}{\mu_{ij}}}$$

T - approximated total sojourn time for the particular job.

Algorithm:

Step 1: Determine $NJOB(j)$ for each shop j according to the following

$$NJOB(j) = \sum_{1 \le i \le j} n_i(0), \qquad j = 1, ...M.$$

where $n_i(0)$ = number of jobs at shop i at time 0. (See Section 3.2.1.), and M = number of shops.

Step 2: Determine the expected shop sojourn time and variance at shop 1, $STAGE(1)$ and $SVAR(1)$ as

$$STAGE(1) = n_1(0)/\hat{\mu_1},$$

$$SVAR(1) = n_1(0)/\sum_{1 \le i \le p_j} \mu_{1j}^2,$$

and set the number of completions at shop 1 to its deterministic value, i.e.,

$$NUMC(1) = n_1(0).$$

Step 3: Determine $STAGE(j)$ and $NUMC(j)$ for all $j = 2, ...M$ as follows:

$$NUMC(j) = MAX(1, NJOB(j) - (\sum_{1 \le i < j} STAGE(i)) * \hat{\mu_j}) \quad (3.9)$$

$$STAGE(j) = NUMC(j)/\hat{\mu_j} \quad (3.10)$$

$$SVAR(j) = NUMC(j)/\sum_{1 \le i \le p_j} \mu_{ij}^2. \quad (3.11)$$

In Equation (3.9) the use the maximum ensures that the number of jobs at shop j is at least one (the particular job itself).

Step 4: Determine Total Sojourn Estimate Mean and Variance as follows:

$$E[T] = \sum_{1 \le j \le M} STAGE(J) \quad (3.12)$$

$$Var[T] = \sum_{1 \le j \le M} SVAR(J). \quad (3.13)$$

The algorithm is coded in a FORTRAN program a listing of which is provided in Appendix B.1.

The reason that the technique is effective lies in the symmetry of the associated distributions. It should be intuitively obvious that using mean values should lead to good estimate of the overall mean even for skewed

distributions. However, for the variance, two key factors affect the performance of such a heuristic. First, for the Markovian systems under consideration, the form of the variance of the individual service times is proportionally additive. That is, the variance associated with N service completions from identical service distributions is N times the variance of a single completion. The second factor is the symmetry involved. These factors allow for averaging of variances across all possible values of N to yield an effective estimate of the true variance.

3.3.3 Enhanced Deterministic Simulation Heuristic (DPL)

An inherent problem with the DSH technique occurs in situations where the service variances at the individual shops are large and, according to the heuristic, the particular job finds no jobs or only a few jobs ahead of it at a given shop. In these cases, it was anticipated that the heuristic would give a low estimate of the mean and variance contribution from that shop to the total sojourn time.

The following paragraphs provide the motivational details for an enhanced version of the deterministic simulation heuristic, which will be referred to as the DPL technique.

Table 3.5: Coefficient of Skewness for some Representative Distributions

Stages of Service	Number in Queue				
	1	2	3	4	5
1	2.12	1.63	1.44	1.34	1.28
2	1.63	1.34	1.23	1.18	1.14
3	1.44	1.23	1.16	1.12	1.10
5	1.28	1.14	1.10	1.07	1.06
10	1.14	1.07	1.05	1.04	1.03

The cases where the particular job finds a near-empty queue upon arrival at a given shop are potentially troublesome because of the skewed shop sojourn-time distributions that result. The coefficient of skewness is displayed in Table 3.5 for the sojourn time distribution at a single shop with some representative special-Erlang service distributions and queue sizes of 1, 2, 3, 4, and 5. From this table it can seem that as the queue size increases and/or the number of phases of service increases (for a constant mean service time, hence tighter variance) the coefficient of skewness decreases. If the queue size is sufficiently large then the skewness coefficient is effectively 1 and hence the sojourn distribution is effectively symmetric. The same is true if the number of phases of service is sufficiently large or if the combination of queue size and number of phases is large enough.

The DSH technique assumes that the sojourn time distribution for the particular job at each shop in the serial network is effectively symmetric. In this case, a deterministic approximation turns out to be a good estimate. However, if the sojourn distribution at a shop is skewed because the combination of queue size and number of phases of service is small then a deterministic approximation will tend to underestimate the true shop sojourn mean. This occurs because of the bottoming out effect that having none or very few jobs queued and a small number of service phases causes. There is obviously a minimum for the contribution to the total sojourn-time from a given shop. This lower bound value approaches zero. Since the exponential and phase-type exponential distributions are theoretically unbounded, the upper bound on the contribution is unbounded. However, practical bounds can be established by using such rules of thumb as the three standard deviation rule.

Therefore, if the mean shop sojourn-time is large enough with respect to the variance of this time then the distribution should be relatively symmetric by the Central Limit Theorem. Due to this symmetry, using only

the mean shop sojourn time and the variance associated with this value should give a good approximation to the actual value.

The DPL heuristic attempts to account for this problem. If the number of queued jobs at a given shop j is large (greater than 3) upon arrival the particular job, then DPL uses the DSH approximation for the mean and variance contributions at that shop. However, if this value is small at shop j, then the estimate for the shop sojourn is adjusted.

The DPL heuristic incorporates a probabilistic estimate for the shop sojourn mean for those troublesome cases just described. This estimate is based on the following observations.

Consider a simple 2 queue network with exponential servers working at rates μ_1 and μ_2, respectively. Assume that both servers are busy. The probability that the first server completes its service before the second is

$$Pr[1] = \frac{\mu_1}{(\mu_1 + \mu_2)}. \tag{3.14}$$

This can be taken directly from Figure 3.2 for any node where both servers are active (node (2,1) for example).

Now if the queue at shop 1 is empty and the queue at shop 2 contains 1 job, and the job in service at shop 1 is considered the particular job, then

the probability that the particular customer completes service and finds a total of 3 jobs at shop 2 is given in Equation 3.14. The probability that it finds 3 jobs, 2 jobs, and 1 job can be derived straight from Figure 3.2 as the path probabilities from node (1,2) to (0,3), (0,2), and (0,1), respectively. Explicitly,

$$p_3 = \frac{\mu_1}{(\mu_1 + \mu_2)}$$

$$p_2 = \frac{\mu_1 \mu_2}{(\mu_1 + \mu_2)^2}$$

$$p_1 = 1 - (p_2 + p_3) = \frac{\mu_2^2}{(\mu_1 + \mu_2)}$$

The exact probabilities for special Erlang service instead of exponential can be computed but involve a much more complicated formula not in keeping with the goal of simple heuristic methodologies. For example, consider the same scenario as just described except assume that the service distributions are 2 stage special Erlang. The three probabilities in this case are

$$p_3 = \frac{\hat{\mu}_1^2(\hat{\mu}_1 + 3\hat{\mu}_2)}{(\hat{\mu}_1 + \hat{\mu}_2)^3}$$

$$p_2 = \frac{3\hat{\mu}_1^3\hat{\mu}_2^2 + 7\hat{\mu}_1^2\hat{\mu}_2^3}{(\hat{\mu}_1 + \hat{\mu}_2)^5}$$

$$p_1 = \frac{5\hat{\mu}_1\hat{\mu}_2^4 + \hat{\mu}_2^5}{(\hat{\mu}_1 + \hat{\mu}_2)^5}$$

where $\hat{\mu}_j$ is the total service rate at shop j.

Note that for nearly equal rates $p_3 \approx 0.5$ which is also true for p_3 for the exponential service case. However, the associated shop sojourn time is not simply a weighted combination of a 3-fold convolution of a 2-stage Erlang, a 2-fold convolution, and a single 2-stage distribution where the weights are the above probabilities. For this to be true the stage of service in progress at shop 2 would have to be stage 1 upon completion of the particular job at shop 1. Thus, the true shop sojourn time would be somewhat less than the weighted combination just described. Determination of the exact distribution is certainly possible. However even for this simple example it would be quite complex.

Therefore, to avoid the complicating factors associated with determining the exact probabilites of the number of jobs the particular job will find at successive shops, the DPL heuristic uses a simplification of the above derivation for those cases where the deterministic approximation of the number of jobs suggests that the value is small. It adjusts the probabilities derived for the exponential case for use in those cases where staged Erlang

service is assumed. The adjustment gives more weight towards the events associated with fewer jobs than the weight derived above for the exponential case. This scheme is in keeping with the goal of a simple, yet accurate, heuristic.

For the following let L represent the number of jobs potentially at shop j upon arrival of the particular job according to the DSH technique. Based upon empirical evidence and the coefficients of skewness shown in Table 3.5 if the number of jobs at a shop upon arrival of the particular job is greater than 3 then the standard DSH estimates are used to approximate the sojourn contribution from that shop. However, if this value is 3 or less then the following is used.

Let

$$P_L = \frac{\hat{\mu}_{j-1}}{\hat{\mu}_j + \hat{\mu}_{j-1}}$$

$$P_l = P_1\left(\frac{\hat{\mu}_j}{\hat{\mu}_j + \hat{\mu}_{j-1}}\right)^{r_j*(L+1-l)}$$

where $l = 2, ..., L-1$ and r_j = number of phases of service at shop j.

$$P_1 = 1.0 - \sum_{l=2}^{l} P_l$$

$$NUMC(j) = \sum_{l=1}^{L} l * P_l$$

The stage variance, $SVAR(j)$, is computed via Equation 3.11 if L is greater than 3, otherwise the following is used:

$$SVAR(j) = \sum_{l=1}^{L} \frac{l * P_l * r_j}{\hat{\mu}_j}$$

The stage sojourn time, $STAGE(j)$, is calculated via Equation 3.10 as in the DSH estimate except the $NUMC(j)$ computed above is used. The expected sojourn time, $E[T]$, and the expected sojourn variance, $Var[T]$, are computed via Equations 3.12 and 3.13.

A listing of the DPL program is provided in Appendix B.2.

3.4 Experimental Design

An experimentation plan was developed in order to evaluate the effectiveness of both the DSH and DPL heuristics versus the true solution or an empirical approximation based upon the Monte-Carlo simulation model. The plan also served to validate the Monte-Carlo model so that it could be used in those cases where the exact solution was unattainable.

The plan consisted of a set of example scenarios which were designed to test the heuristics under conditions similar to those found in the NARF-like facilities described in Chapter 1. The dominating factor of these facilities is heavy traffic. Scenarios which did not fall into this category were also included to test the robustness of the heuristics.

The experimentation was designed around four key parameters: the number of shops, the initial queue lengths, the shop service rates, and the service distributions. The number of shops was set at four levels; $2, 3, 5$, and 10. The queue lengths were varied from 1 to 12.

The service rates were normalized to a base value of 1.0. For purposes of emulating NARF-like facilities, rates between 0.8 and 1.0 were used. However, as low as 0.2 were also included to test robustness.

A set of examples was developed to test a full design of the levels of each of these three factors. These examples are displayed in Table 3.6, Table 3.7, Table 3.8, and Table 3.9. Each example was solved for each of 5 different service time distributions. These distributions included the exponential distribution; three special Erlang distributions where the number of stages took on the values $2, 3$, and 5; and a truncated normal distribution where

Table 3.6: Example Numbers for 2 Queue Scenarios

Example Number	Service Rate at Shops 1 - 2						
Queue Size at Shop	1.0	0.8	0.5	0.2	1.0	1.0	1.0
1, 2	1.0	1.0	1.0	1.0	0.8	0.5	0.2
1,1	1	26	51	76	101	126	151
1,2	2	27	52	77	102	127	152
1,5	3	28	53	78	103	128	153
1,8	4	29	54	79	104	129	154
1,12	5	30	55	80	105	130	155
2,1	6	31	56	81	106	131	156
2,2	7	32	57	82	107	132	157
2,5	8	33	58	83	108	133	158
2,8	9	34	59	84	109	134	159
2,12	10	35	60	85	110	135	160
5,1	11	36	61	86	111	136	161
5,2	12	37	62	87	112	137	162
5,5	13	38	63	88	113	138	163
5,8	14	39	64	89	114	139	164
5,12	15	40	65	90	115	140	165
8,1	16	41	66	91	116	141	166
8,2	17	42	67	92	117	142	167
8,5	18	43	68	93	118	143	168
8,8	19	44	69	94	119	144	169
8,12	20	45	70	95	120	145	170
12,1	21	46	71	96	121	146	171
12,2	22	47	72	97	122	147	172
12,5	23	48	73	98	123	148	173
12,8	24	49	74	99	124	149	174
12,12	25	50	75	100	125	150	175

Table 3.7: Example Numbers for 3 Queue Scenarios

Example Number	Service Rate at Shops 1 - 3						
	1.0	1.0	1.0	0.8	1.0	0.8	0.8
Queue Size at Shop	1.0	1.0	0.8	1.0	0.8	1.0	0.8
1, 2, 3	1.0	0.8	1.0	1.0	0.8	0.8	1.0
1,1,1	176	204	232	260	288	316	344
5,1,1	177	205	233	261	289	317	345
1,5,1	178	206	234	262	290	318	346
1,1,5	179	207	235	263	291	319	347
5,5,1	180	208	236	264	292	320	348
5,1,5	181	209	237	265	293	321	349
1,5,5	182	210	238	266	294	322	350
5,5,5	183	211	239	267	295	323	351
8,1,1	184	212	240	268	296	324	352
1,8,1	185	213	241	269	297	325	353
1,1,8	186	214	242	270	298	326	354
8,8,1	187	215	243	271	299	327	355
8,1,8	188	216	244	272	300	328	356
1,8,8	189	217	245	273	301	329	357
8,5,5	190	218	246	274	302	330	358
5,8,5	191	219	247	275	303	331	359
5,5,8	192	220	248	276	304	332	360
8,8,5	193	221	249	277	305	333	361
8,5,8	194	222	250	278	306	334	362
5,8,8	195	223	251	279	307	335	363
8,8,8	196	224	252	280	308	336	364
7,7,7	197	225	253	281	309	337	365
7,8,8	198	226	254	282	310	338	366
8,7,8	199	227	255	283	311	339	367
8,8,7	200	228	256	284	312	340	368
7,7,8	201	229	257	285	313	341	369
7,8,7	202	230	258	286	314	342	370
8,7,7	203	231	259	287	315	343	371

Table 3.8: Example Numbers for 5 Queue Scenarios

Example Number	Service Rate at Shops 1 - 5						
	1.0	0.8	1.0	1.0	0.8	0.9	0.85
	1.0	0.85	0.95	0.9	0.9	1.0	0.9
	1.0	0.9	0.9	0.8	1.0	0.9	0.8
Queue Size at Shop	1.0	0.95	0.85	0.9	0.9	1.0	0.95
1, 2, 3, 4, 5	1.0	1.0	0.8	1.0	0.8	0.9	1.0
10,10,10,10,10	373	388	404	420	436	452	468
10,8,9,10,7	373	389	405	421	437	453	469
7,8,9,10,10	374	390	406	422	438	454	470
10,9,8,9,10	375	391	407	423	439	455	471
8,9,10,9,8	376	392	408	424	440	456	472
7,10,5,8,9	377	393	409	425	441	457	473
6,10,9,7,8	378	394	410	426	442	458	474
10,9,10,8,10	379	395	411	427	443	459	475
8,8,8,10,10	380	396	412	428	444	460	476
10,10,6,9,8	381	397	413	429	445	461	477
9,10,9,8,7	382	398	414	430	446	462	478
10,10,9,9,9	383	399	415	431	447	463	479
9,7,8,9,10	384	400	416	432	448	464	480
5,5,5,5,5	385	401	417	433	449	465	481
5,6,7,7,6	386	402	418	434	450	466	482
9,9,10,10,9	387	403	419	435	451	467	483

Table 3.9: Example Numbers for 10 Queue Scenarios

Example Number	Service Rate at Shops 1 - 10						
	1.0	1.0	0.8	1.0	0.8	0.8	0.7
	1.0	0.9	0.85	0.95	0.9	1.0	1.0
	1.0	1.0	0.9	0.9	1.0	0.9	0.8
	1.0	0.9	0.95	0.85	0.9	1.0	0.9
	1.0	1.0	1.0	0.8	0.8	0.8	0.8
	1.0	0.9	0.95	0.8	0.8	1.0	1.0
	1.0	1.0	0.9	0.85	0.9	0.9	0.9
	1.0	0.9	0.85	0.9	1.0	1.0	0.9
Queue Size at Shop	1.0	1.0	0.8	0.95	0.9	0.8	0.8
1, 2, 3, 4, 5, 6, 7, 8, 9, 10	1.0	0.9	0.8	1.0	0.8	1.0	1.0
10,10,10,10,10,10,10,10,10,10	484	500	516	532	548	564	580
8,9,10,8,10,9,8,10,10,9	485	501	517	533	549	565	581
10,9,10,10,9,7,8,9,8,7	486	502	518	534	550	566	582
8,8,8,9,9,9,10,10,10,10	487	503	519	535	551	567	583
10,10,10,9,9,9,8,8,8,8	488	504	520	536	552	568	584
5,5,6,7,7,8,9,9,10,10	489	505	521	537	553	569	585
10,5,10,5,10,5,10,5,10,5	490	506	522	538	554	570	586
10,8,10,8,10,8,10,8,10,8	491	507	523	539	555	571	587
10,5,9,6,10,7,8,6,9,8	492	508	524	540	556	572	588
5,6,7,10,10,10,6,5,8,7	493	509	525	541	557	573	589
6,7,8,9,10,10,9,8,7,6	494	510	526	542	558	574	590
5,5,5,6,10,6,5,6,5,10	495	511	527	543	559	575	591
10,10,5,6,8,10,9,7,8,6	496	512	528	544	560	576	592
8,7,9,10,7,10,8,10,9,9	497	513	529	545	561	577	593
10,10,10,9,8,8,9,10,10,10	498	514	530	546	562	578	594
5,6,7,8,9,10,10,9,10,8	499	515	531	547	563	579	595

the variance corresponded to an equivalent 10 stage Erlang. These distributions provided a means to determine the effect of the variance on the heuristics. The exponential distribution case provided the highest variance case and the variance was decreased in the 2 stage Erlang, and so on with the normal distribution providing the lowest variance case. Henceforth, the example datasets with the exponential service times are referred to as the Model 1 datasets, those with 2 stage Erlang distribution are Model 11, those with 3 stage are Model 21, those with 5 stage are Model 31, and finally those with the truncated normal are Model 41.

3.5 Results

In this section the results of the experimentation plan are presented. This presentation is organized as follows. First, computational requirements of the exact technique are presented in Section 3.5.1. Next, the results of the validation of the Monte-Carlo simulation runs are provided in Section 3.5.2. Finally, the results and analysis of the performance of the DSH and DPL heuristic techniques are given in Section 3.5.3.

3.5.1 Computational Requirements of Exact Solution

The exact analytic algorithm was implemented for those Model 1 example scenarios with 2 queues in series. Attempts to solve larger problems were aborted due to computational storage and time requirements that were beyond acceptable limits. Approximately 10 megabytes of core storage were required for the 2 queue problems with maximum of 30 total jobs in the system. It should be noted that a scheme to reduce the storage requirements was implemented. This scheme was based on the observation that state-space paths may have duplicate traversal distributions and thus have identical terms in the overall sojourn distribution. This duplicity is fairly common especially in systems with nearly equal initial queue lengths and can reduce the storage requirements by as much as a couple of orders of magnitude.

One problem with this is that the amount of duplicity is dependent upon the shape of the state space and thus is very difficult, if not impossible, to determine a priori. Another observation led to further reduction of storage. This scheme takes advantage of the fact that every state is one-step reachable by at most M other states where M is the number of queues.

Thus, during the back substitution process once a state has been used M times all information stored about that state can be eliminated, thus freeing the associated storage locations.

However, even using these schemes, small 3 queue examples required more storage than was available (18 megabytes). The biggest contributing factor to the space requirements is the number of possible paths through the state space. Even relatively efficient schemes to transfer core storage requirements to high speed disks were prohibitive given the storage required.

3.5.2 Monte-Carlo Simulation Validation and Calibration

The Monte-Carlo simulation model was developed both for evaluation of its own merits as a tool for solving the problem at hand and to provide a reliable approximation of the true sojourn-time distributions.

In order to use this technique the model results were first compared with the exact results to ensure that the model was correctly defined. This comparison also provided a means for calibrating the number of iterations that were required to provide results within given tolerance levels and yet minimize the computer time required to run the simulation.

The validation process involved performing statistical tests on the empirical mean and variance from the simulation versus the exact values from the analytic model. In addition, distribution tests were also performed. Each of these are discussed in detail below.

A simple two-tailed statistical hypothesis test on the mean from the simulation versus the exact mean for the 2 queue Model 1 scenarios (examples 1-175) was performed. The hypothesis test was as follows:

$$H_0 : \mu = \mu_0$$

$$H_1 : \mu \neq \mu_0$$

where

μ = mean from simulation model, and

μ_0 = exact mean.

The test statistic, Z, was

$$Z = \frac{\mu - \mu_0}{\sigma/\sqrt{n}},$$

where σ = true standard deviation. The acceptance region for this test was

$$-z_{\alpha/2} < Z < z_{\alpha/2}$$

where α is the probability of rejecting the null hypothesis H_0 when it is in fact true.

As with the mean, a simple two-tailed test on the variance was also performed. In particular, a chi-square statistic was used as follows:

$$H_0 : \sigma^2 = \sigma_0^2$$

$$H_1 : \sigma^2 \neq \sigma_0^2$$

where σ^2 = variance from simulation model, and σ_0^2 = exact variance.

The test statistic, X^2, was

$$X^2 = \frac{(n-1)\sigma^2}{\sigma_0^2}$$

$\gamma = n - 1$ degrees of freedom

n = number of iterations of the simulation.

The acceptance region was

$$\chi^2_{\frac{\alpha}{2}} < X^2 < \chi^2_{1-\frac{\alpha}{2}}.$$

The distributions of sojourn times built empirically from the iterations of the Monte-Carlo model were compared to the distributions found via

the exact technique. A Kolmogorov-Smirnov test was employed as means of determining the goodness of fit. However, due to the form of the exact solution, determination of interval probabilities would have required explicit reinversion from the transform domain to the time domain. The large number of terms in the transform made this an unattractive computational option. Instead, the previously mentioned heuristic developed by Platzmon, Ammons, and Bartholdi was used to approximate the interval probabilities directly from the transform solution.

Based upon the results of these tests, it was found that 5000 iterations of the Monte-Carlo model more than ensured that none of the null hypotheses were rejected at the 0.05 level for any of the 2 queue examples from the Model 1 dataset. This value was then used for all further Monte-Carlo runs for all example datasets.

3.5.3 Results of Heuristics

In this section the results of the heuristic DSH and DPL techniques are compared to those of the exact solution and the Monte-Carlo simulation model. These results are broken down by mean comparison and standard deviation comparison.

Comparison of means. The mean sojourn-times from the two heuristics were first compared with those from the exact technique. For purposes of comparison the percentage difference between the approximated mean sojourn time from both heuristics and the exact solution was computed. The actual value of the sojourn means and standard deviations are of no importance since they are relative to the normalized service time distributions.

The percentage differences for each 2 queue example scenario in the Model 1 dataset are displayed in Appendix C.1. Comparisons with larger problems was not attainable because of the computational requirements of the exact technique. Table 3.10 summarizes the results of Appendix C.1. Notice that the DPL heuristic provided better results than the DSH heuristic for these small example scenarios. Notice also that on the average both techniques tended to underestimate the true mean.

The computational requirements of the exact technique prohibited the solving of larger problems. Therefore, the Monte-Carlo simulation model was employed for comparison purposes for all datasets. Appendices C.2 - C.3 show the percentage differences between the means from two heuristics

Table 3.10: Summary of Percentage Differences of the Estimated Sojourn Means from the DSH and DPL Techniques vs. the Exact Analytic Solution

For Model Number 1 For Examples 1-175		
	DSH	DPL
Average Absolute % Difference	4.806	2.475
Average % Difference	4.806	1.458
Standard Deviation of the % Difference	6.211	3.599

and the simulation model for selected models and queue sizes. The results from all scenarios are summarized by average absolute difference, average difference, and standard deviation of the average difference in Tables 3.11-3.13, respectively.

It should be noted from Table 3.12 that the tendency for DSH technique to underestimate the true mean continued for the entire dataset, while the DPL both under-estimated and over-estimated. The tendency for the DSH technique to underestimate the mean is attributable to the skewed shop sojourn times illustrated in Table 3.5. The DSH technique uses the estimate of the number of jobs at a given shop to estimate the associated shop sojourn time. Thus, if it estimated that, for example, only one or two

Table 3.11: Average Absolute Percent Difference for the Mean

Model Number	Heuristic	Number of Queues			
		2	3	5	10
1	DSH	4.91	7.96	4.65	6.08
	DPL	2.64	5.80	4.63	4.26
11	DSH	3.00	4.93	1.83	2.34
	DPL	1.76	3.30	1.14	1.67
21	DSH	2.20	3.71	1.01	1.41
	DPL	1.94	3.18	1.00	1.28
31	DSH	1.47	2.60	0.45	0.74
	DPL	1.56	2.37	0.37	1.24
41	DSH	0.91	1.66	0.16	0.29
	DPL	1.64	2.47	0.17	1.31

Table 3.12: Average Percent Difference for the Mean

Model Number	Heuristic	Number of Queues			
		2	3	5	10
1	DSH	4.80	7.95	4.45	5.22
	DPL	1.46	2.52	-0.48	-0.30
11	DSH	2.90	4.88	1.82	2.34
	DPL	0.34	1.13	0.98	-0.16
21	DSH	2.09	3.65	0.99	1.40
	DPL	1.83	2.94	0.99	1.21
31	DSH	1.36	2.54	0.41	0.73
	DPL	-0.87	-0.96	0.29	-0.86
41	DSH	0.81	1.61	0.13	0.28
	DPL	-1.36	-1.95	0.04	-1.22

jobs (including the particular job) will be at a shop when the particular job arrives, the DSH approximation will underestimate the true shop sojourn mean since the lower bound on the number of jobs at the shop is one (the particular job itself). The true value in such a case will always be greater than the estimated value because there is a finite probability that the actual number of jobs at the shop could be as large as the total number of jobs initially at that shop and all previous shops. (See the definition of NJOB(j) in Section 3.3.2.)The DPL technique attempts to account for this bias as described earlier in Section 3.3.3. This is evidenced in Tables 3.11 and 3.12.

Notice from Table 3.11 that the DPL technique outperforms the DSH heuristic in the high variance models (1, 11, and 21), while the simpler DSH techique is slightly better in the low variance models (31 and 41). Also, notice that the estimated means from DSH and DPL both tend to get better as the variance gets smaller. However, the DSH technique shows a more dramatic reduction. For example, for the 2 queue examples, the DSH mean ranged from a worse case average absolute difference of 4.91% for the Model 1 examples to a difference of 0.91% for the Model 41 examples. For the DPL heuristic, this difference ranged from a worse case of 2.64%

for the Model 1 examples down to a difference of 1.56% for the Model 31 examples.

This same kind of pattern exists for the 3, 5, and 10 queue scenarios. It should be noted that comparisons of values in this table across queue sizes must be done carefully since the examples do not reflect a consistency in the queue service rate ratios. However, individual examples where the queue sizes and service rates exhibit similar patterns across 2, 3, 4, and 10 queue problems can be compared. For instance, example numbers 19, 196, 373, and 499 all have service rates equal to 1.0 at each shop with large, equal queue lengths (8 for examples 19 and 196 and 10 for examples 373 and 499). Table 3.14 summarizes the results for these examples across all models for the DSH technique. The results are nearly identical for the DPL heuristic.

Notice the general tendency for better estimates as the shop service variance tightens. Notice, also, that the estimate of the overall sojourn mean gets worse as the number of queues increases. However, for the tightened service variance cases (Models 31 and 41) this is almost neglible.

Table 3.13: Standard Deviation of the Average Percent Difference for the Mean

Model Number	Heuristic	Number of Queues			
		2	3	5	10
1	DSH	6.15	7.88	3.02	4.51
	DPL	3.56	4.29	3.75	4.73
11	DSH	4.47	6.17	1.69	1.78
	DPL	2.39	3.53	1.32	1.16
21	DSH	3.62	5.20	1.15	1.27
	DPL	3.09	4.13	1.13	1.27
31	DSH	2.76	4.06	0.66	0.81
	DPL	2.54	3.82	0.50	1.03
41	DSH	1.95	2.91	0.32	0.42
	DPL	3.02	4.72	0.28	1.19

Table 3.14: Absolute Percent Difference in the Mean for DSH for Some Particular Examples

Percent Difference	Example Number			
Model Number	19	196	373	499
1	1.17	2.40	5.24	5.50
11	0.59	0.83	2.16	2.49
21	0.31	0.46	1.12	1.43
31	0.29	0.01	0.54	0.63
41	0.14	0.13	0.06	0.12

The standard deviation of the average difference in the means for all models and queue sizes is displayed in Table 3.13. These values demonstrate that both techniques produce relatively stable estimates for the mean sojourn time across a wide range of scenarios. Note that the DSH estimates ranged from a worse case of 7.88% down to 0.32% while the corresponding range for the DPL estimates is 4.73% to 0.28%.

Comparison of standard deviations. As in the previous section the standard deviation estimates from the heuristics were first compared with that from the exact distribution for each 2 queue Model 1 scenarios.

Appendix C.4 compares of the standard deviations from the DSH and DPL heuristics versus the exact standard deviation for the examples solved. These results are summarized in Table 3.15.

The standard deviation from the heuristic was also compared with that from the Monte-Carlo simulation for the entire dataset. The comparisons for selected models and queue sizes are displayed in Appendices C.5 - C.6. The results for all scenarios are summarized by average absolute difference, average difference, and standard deviation of the average difference in Tables 3.16 - 3.18.

Table 3.15: Summary of Percentage Differences of the Estimated Sojourn Standard Deviations from the DSH and DPL Techniques vs. the Exact Analytic Solution

For Model Number 1 For Examples 1-175		
	DSH	DPL
Average Absolute % Difference	5.874	6.611
Average % Difference	-2.528	-3.481
Standard Deviation of the % Difference	6.980	7.068

Table 3.16: Average Absolute Percent Difference for the Standard Deviation

Model Number	Heuristic	Number of Queues			
		2	3	5	10
1	DSH	6.03	6.78	13.64	17.49
	DPL	6.80	8.21	13.50	17.02
11	DSH	6.56	6.75	10.88	12.53
	DPL	7.47	8.44	10.72	11.99
21	DSH	6.93	6.57	9.74	9.83
	DPL	7.54	6.66	9.46	8.82
31	DSH	7.02	6.56	7.99	7.50
	DPL	7.96	8.13	7.82	6.96
41	DSH	7.86	7.63	6.44	5.78
	DPL	8.82	9.24	6.28	5.54

Table 3.17: Average Percent Difference for the Standard Deviation

Model Number	Heuristic	Number of Queues			
		2	3	5	10
1	DSH	-2.55	-5.53	-13.49	-17.43
	DPL	-3.51	-7.37	-13.35	-16.96
11	DSH	-3.13	-5.42	-10.36	-12.43
	DPL	-3.93	-7.14	-10.21	-11.86
21	DSH	-3.22	-5.18	-8.43	-9.59
	DPL	-2.43	-5.07	-8.15	-8.40
31	DSH	-3.07	-4.71	-5.76	-6.21
	DPL	-3.76	-6.32	-5.59	-5.58
41	DSH	-3.91	-5.17	-2.59	-3.04
	DPL	-4.60	-6.85	-2.43	-2.42

Table 3.18: Standard Deviation of the Average Percent Difference for the Standard Deviation

Model Number	Heuristic	Number of Queues			
		2	3	5	10
1	DSH	6.87	5.55	8.34	8.80
	DPL	6.97	5.63	8.20	8.39
11	DSH	8.67	5.96	9.17	10.57
	DPL	8.81	6.45	8.97	9.74
21	DSH	9.79	6.06	8.65	10.68
	DPL	9.85	5.91	8.26	9.66
31	DSH	10.52	5.75	6.98	9.73
	DPL	10.67	6.96	6.74	8.78
41	DSH	12.41	7.01	4.78	8.51
	DPL	12.86	9.33	4.57	7.46

From the percent deviation of the standard deviation estimates shown in Table 3.17 it can be seen that both of the techniques tend to overestimate the total sojourn variance. This is understantable given the negative correlation that exists between individual shop sojourn times. As mentioned previously, neither heuristic attempted to account for this correlation due to the complex nature of this phenomenon.

On the average, the DSH technique provides better estimates than the DPL heuristic for the smaller networks while the opposite is true for the larger networks. However, there is not a significant difference between the two.

Refering back to Table 3.16 other tendencies besides simply overestimation of the true standard deviation of the sojourn time can be seen. For example, in the higher service variance cases (Models 1, 11, and 21). Estimates for the 2 queue examples get slightly worse as the shop service variances tighten. However, as the number of queues in series increase, this tendency is reversed with the most dramatic reduction occurring in the 10 queue examples (down from 17.49% to 5.78% for the DSH technique and 17.02% to 5.67% for the DPL).

As pointed out previously, there is inconsistency in the relative service rates in the examples scenarios going across the table (increasing number of queues). This makes comparisons across the table meaningless. It can be noted from Table 3.16 that in the high variance models (1 and 11) the estimates get worse as the number of queues in series increase. However, in the low variance models (21, 31, and 41) the difference as the number of queues increases is neglible and in fact in the Model 41 examples the estimates get slightly better (down from 7.86% to 5.78% for the DSH technique and 8.90% to 5.67% for the DPL).

This is further evidenced by the comparing the absolute difference in the standard deviation for the particular examples sited in the previous subsection for which the service rates and queue sizes are consistent across the network sizes. These values are displayed in Table 3.19.

The standard deviation of the difference in the estimated sojourn standard deviation for each model and network size are displayed in Table 3.18. This table suggests that both techniques are fairly stable in predicting the sojourn variance across the wide range of example scenarios. This is

Table 3.19: Absolute Percent Difference in the Standard Deviation for DSH for Some Particular Examples

Percent Difference	Example Number			
Model Number	19	196	373	499
1	5.91	6.14	12.93	20.19
11	0.82	5.08	12.50	15.24
21	0.10	1.52	8.78	13.40
31	1.36	1.42	4.82	11.13
41	0.23	1.06	1.38	3.68

evidenced by a worse case value of 12.41% for the DSH technique and 12.96% for the DPL. In general, these values lie between 4% and 10%.

3.6 Conclusions

This chapter has been concerned with the solution of sojourn-time distributions for particular jobs in serial networks of queues.

The emphasis was the attainment of such distributions in a computationally efficient manner for use in real-time or near real-time applications. An exact technique was developed but shown to be computationally prohibitive. The failings of the exact technique led to the development of heuristic methodologies which could provide a computationally more attractive alternative than straightforward Monte-Carlo simulation.

Two simple procedures were developed which were founded in the Central Limit Theorem. Both of these techniques were shown to give good approximations for the mean and variance of the desired distribution for a range of scenarios. In particular, they were very good for the cases which closely resembled the heavy traffic operation of NARF-like facilities. Both techniques are computationally efficient, requiring approximately 0.05 CPU seconds on a VAX 11/750 running VAX-11 FORTRAN-77 Version 4.

The enhanced heuristic, DPL, was shown to give better estimates of the mean in the high variance scenarios while the simpler DSH technique was slightly better in the low variance scenarios. The techniques performed the same in estimating the sojourn variance. Both techniques tended to slightly overestimate the true sojourn variance due to the inherent correlation that exists between successive shops in the serial network. This correlation is negative which means that an especially long sojourn-time with respect to the average at one shop leads to shorter times at the succeeding shop, while a short sojourn time leads to a longer stay at the succeeding shop. This negative correlation causes the variance of the overall sojourn time to obviously be less than the sum of the individual shop sojourn variances.

CHAPTER 4

DISASSEMBLY-REASSEMBLY NETWORKS

4.1 Introduction

As mentioned in the introduction of this dissertation, the purpose of this research was to investigate two aspects of queueing network problems. Chapter 3 addressed one of these aspects, that is, the determination of sojourn distributions for particular jobs in serial queueing networks. In that chapter it was shown that in practice this is an extremely difficult problem to solve analytically even for very simple serial networks. As a result, in an effort to provide an alternative to Monte-Carlo simulation, simple heuristic schemes were devised and shown to be accurate and robust for a wide range of scenarios.

The second aspect of queueing network problems that this dissertation addresses is that of jobs that require disassembly yielding potentially many subjobs. These subjobs exhibit the same features of a traditional job in

that they require servicing at various shops. However, the subjobs must be reassembled into the parent job upon completion of their subroutes.

This chapter addresses these types of jobs and enhances the heuristics presented in Chapter 3 for problems exhibiting these disassembly-reassembly jobs. The model is explicitly described in Section 4.2. Next, in Section 4.3, the heuristics are developed, with the experimentation plan presented in Section 4.4. Results and analysis are provided in Section 4.5 and, finally, conclusions are offered in Section 4.6.

4.2 Model Description

Within this chapter the following model is considered:

The network consists of N shops. These shops are arranged in the form shown in Figure 4.1. It should be noticed that the network consists of two parallel serial networks sharing common beginning and ending shops. Each shop in the network consists of a FIFO queue and a single server. The service distribution at shop j is generalized Erlang with rate μ_j for all $j = 1, ..., N$. Jobs of a single type arrive at shop 1 where they are

disassembled into two subjobs. One of the subjobs proceeds to the upper serial path and the other to the lower.

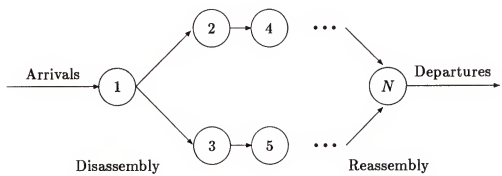


Figure 4.1: Disassembly-Reassembly Network of Single-Server Queues

The subjobs receive service at each shop along their respective subroutes until they join the queue at shop N. Elements in this queue include individual subjobs whose mate has not yet arrived and pairs of mated subjobs. Upon completion of a service operation the server at shop N either begins service to reassemble the top matched pair of subjobs or becomes idle until a matched pair exists.

Given the model as described, the measure of performance of the system that is of interest is the sojourn time distribution for particular jobs under "given-scenario" conditions (as in Chapter 3).

It is worth noting some of the inherent features of this model. First, no job can be overtaken by a later arriving job. This is because the sequence of subjobs arriving to shop N from each serial path must be the same sequence as jobs arriving to the model since each path is simply a serial network of single-server FIFO queues. This means that matched subjob pairs are generated in this same sequence as their parents arrival to shop 1 and therefore reassembled jobs leave in the same sequence.

Secondly, the sojourn time of any job consists of queueing and service time at shop 1, queueing and service time at shop N, and the maximum of the times for each subjob to traverse its respective serial path after leaving shop 1 and before arriving at shop N.

The last element presents an interesting problem, specifically the determination of the maximum of two potentially complex distributions. N.P. Rao [55] approached this problem in the context of two-stage production systems. Rao was interested in determining the cycle time distribution for

tandem production stages with no interstage inventory. (See Figure 4.2 below.) In particular, he was interested in service distributions with tighter variances than that of the exponential which are more accurately approximated as a normal distribution with low σ or as an Erlang with a large number of stages.

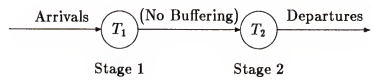


Figure 4.2: Tandem Network of Single-Server Production Stages

For the tandem system under consideration, Rao assumed that the first stage is never starved, the second is never blocked, and that the stage service times are statistically independent. Under this scenario, the cycle time is simply the maximum of the service times at stage 1 and 2. Explicitly, if T_i is a random variable which represents to service time at stage i (i=1,2) and $\mathcal{T} = \text{MAX}[T_1, T_2]$ is a random variable representing the cycle time, then

the cycle time density function, $g_T(\tau)$, is

$$g_T(\tau) = g_1(\tau) \int_0^\tau g_2(t_2)dt_2 + g_2(\tau) \int_0^\tau g_1(t_1)dt_1 \tag{4.1}$$

where

$g_i(\tau)$ = service density function at stage i.

Rao uses Equation 4.1 to derive the mean cycle times for some particular cases where the service distributions are Erlang and/or (truncated) Normal. The derivation of the cycle time distribution from Equation 4.1 is, in general, a difficult task if the stage density functions are complex (such as a generalized hyper-Erlang derived in Chapter 3) and it is, in fact, intractible for certain distributions.

Equation 4.1 can be used to derive the solution to the sojourn distribution determination problem for the disassembly-reassembly network case for certain forms of the individual shop service distributions which result in a Markovian state space. However, given the results of Chapter 3 it is not a computationally attractive alternative.

For the problem at hand, determination of the time for the two subjobs to complete their respective subroutes and rejoin in the queue at shop N certainly involves complicated distributions as evidenced by the exact

results in the previous chapter. From those results it is known that the sojourn distribution along a serial network of Erlang servers is generalized hyper-Erlang. Solution of Equation 4.1 for two such distributions while theoretically possible is not computationally feasible for all but very small problems.

Nonetheless, Equation 4.1 can still be solved for some small problems to gain insight into the form of the distribution that results from taking the maximum of two subroutes. Consider the case where each subroute sojourn time, T_i, has a special Erlang distribution. In particular, the subroute sojourn density function has the following form:

$$g_i(t_i) = \frac{\lambda_i^{n_i}}{(n_i - 1)!} e^{-\lambda_i t_i},$$

where λ_i is the stage service rate of subroute i, and
n_i is the number of stages of service of subroute i.

Let $\mathcal{T}$ represent the maximum of the subroute sojourns, ie., $\mathcal{T} = \mathrm{MAX}[T_1, T_2]$. The resulting density function of $\mathcal{T}$ can be derived directly using Equation 4.1. The characteristics of the distribution of $\mathcal{T}$ are worth noting. In particular, it is interesting to observe the behavior of the mean and variance of $\mathcal{T}$ under various forms of the special Erlang subroute sojourn

times. The case is considered where the individual subroute sojourn times have the same mean but have possibly different stage service rates, λ_i, and number of stages, n_i. The mean and variance of the distribution of $\mathcal{T}$ for some representative λ_i and n_i values are displayed in Tables 4.1 and 4.2, respectively. The mean of the T_i's for these tables is 10.0. The variance of the individual T_i's are displayed in Table 4.3.

In Table 4.1 it can be seen that as the number of stages of service increases (and thus the variance decreases) in either subroute the $E[\mathcal{T}]$ decreases. Notice that since the subroute means are both 10.0 that the $E[\mathcal{T}]$ is in the worst case (both $n_i = 1$) 50% larger than the subroute means. However, it should be noted that for problems where the system is reasonably loaded that the n_i values will not be small. Therefore, the lower right portion of the table is of more interest. This portion indicates that the $E[\mathcal{T}]$ will be within about 15% of the subroute mean.

Considering Table 4.2 and 4.3 it should be noted that with exception of the exponential case (n_i=1) the $\mathrm{Var}[\mathcal{T}]$ is less than the larger of the individual subroute $\mathrm{Var}[T_i]$. Concentrating on the lower right hand portion of Table 4.2 (for the heavier traffic scenarios) it can be seen that the $\mathrm{Var}[\mathcal{T}]$

Table 4.1: Mean of the MAX$[T_1, T_2]$

	n_2									
n_1	1	3	5	10	15	20	30	40	50	75
1	15.0	14.2	14.0	13.9	13.8	13.8	13.7	13.7	13.7	13.7
3	14.2	13.1	12.8	12.5	12.5	12.4	12.3	12.3	12.3	12.3
5	14.0	12.8	12.5	12.1	12.0	12.0	11.9	11.9	11.8	11.8
10	13.9	12.5	12.1	11.8	11.6	11.5	11.4	11.4	11.4	11.3
15	13.8	12.5	12.0	11.6	11.4	11.4	11.3	11.2	11.2	11.1
20	13.8	12.4	12.0	11.5	11.4	11.3	11.1	11.1	11.1	11.0
30	13.7	12.3	11.9	11.4	11.3	11.1	11.0	11.0	10.9	10.9
40	13.7	12.3	11.9	11.4	11.2	11.1	11.0	10.9	10.8	10.8
50	13.7	12.3	11.8	11.4	11.2	11.1	10.9	10.8	10.8	10.7
75	13.7	12.3	11.8	11.3	11.1	11.0	10.9	10.8	10.7	10.7

See Table 4.3 for corresponding λ_i values.

Table 4.2: Variance of the MAX$[T_1, T_2]$

	n_2									
n_1	1	3	5	10	15	20	30	40	50	75
1	125.0	78.8	70.8	65.2	63.5	62.6	61.7	61.3	61.0	60.7
3	78.8	34.0	26.4	21.2	19.6	18.8	18.0	17.6	17.4	17.1
5	70.8	26.4	18.9	13.7	12.2	11.4	10.6	10.3	10.1	9.8
10	65.2	21.2	13.7	8.7	7.1	6.3	5.6	5.3	5.1	4.8
15	63.5	19.6	12.2	7.1	5.5	4.8	4.1	3.7	3.5	3.3
20	62.6	18.8	11.4	6.3	4.8	4.1	3.3	3.0	2.8	2.5
30	61.7	18.0	10.6	5.6	4.1	3.3	2.6	2.3	2.1	1.8
40	61.3	17.6	10.3	5.3	3.7	3.0	2.3	1.9	1.7	1.5
50	61.0	17.4	10.1	5.1	3.5	2.8	2.1	1.7	1.5	1.3
75	60.7	17.1	9.8	4.8	3.3	2.5	1.8	1.5	1.3	1.0

See Table 4.3 for corresponding λ_i values.

Table 4.3: Mean and Variance of Some Representative T_i

Stage Rate	Number of Stages	Mean	Variance
0.1	1	10.0	100.0
0.3	3	10.0	33.3
0.5	5	10.0	20.0
1.0	10	10.0	10.0
1.5	15	10.0	6.7
2.0	20	10.0	5.0
3.0	30	10.0	3.3
4.0	40	10.0	2.5
5.0	50	10.0	2.0
7.5	75	10.0	1.3

is more than 15% less than the larger of the subroute variances. This value is more than 30% for the highest values of n_i (ie., lowest variance) when $n_1 = n_2$. Note also that for the cases where one of the subroute variances is particularly smaller than other, the corresponding $\mathrm{Var}[\mathcal{T}]$ may be more than 50% smaller than the larger of the subroute variances.

Heuristic methodologies for approximating the sojourn mean and variance of $\mathcal{T}$ have been developed which incorporate many of the observations just described. These are presented in the next section.

4.3 Heuristic Solution Procedures

As in Chapter 3, heuristic methodologies have been developed as approximation procedures for the sojourn mean and variance for particular customers traversing the disassembly-reassembly networks just described. The key consideration in the development of the heuristics was the goal of relatively accurate, yet, simple schemes that are computationally feasible in near real-time applications. In this section four such techniques are presented.

4.3.1 Deterministic Simulation Heuristic (DSH)

The first heuristic considered was the DSH technique described in Chapter 3. This scheme assumes all service times are deterministic and equal to the mean of the actual service distribution. Under this technique the sojourn time associated with the subjob serial routes is simply the maximum of the two deterministically computed subroute sojourn times. The sojourn variance is computed by summing variances according to the number of services the particular job sees at each shop (which is equal to the number of jobs at each shop upon arrival of the particular subjob, including the particular subjob) in the longest deterministic path through the disassembly-reassembly network. See Section 3.3.2 for details of the algorithm for the DSH technique and Appendix B.1 for a listing of the FORTRAN program.

4.3.2 Enhanced Deterministic Simulation Heuristic (DPL)

The DPL heuristic from Chapter 3 was also employed for the disassembly-reassembly networks. As with the DSH technique just described it determines the longest of the two subroute serial paths using deterministic service times. The DPL technique described in Section 3.3.3 is then performed

on the resulting serial network yielding mean and standard deviation estimates for the sojourn times. The FORTRAN program listing of the DPL heuristic is provided in Appendix B.2.

4.3.3 Deterministic Simulation Heuristic with Subroute Maximization (DSHSM)

Based upon the results displayed in Tables 4.1 - 4.3 and intuition it can be seen that the mean of the maximum of two distributions is greater than that of either of the individual distributions while the variance of the maximum is, in general, less than the larger of the individual variances. These observations, coupled with the general tendency of the DSH heuristic to underestimate the mean and overestimate the variance of the serial models in Chapter 3, indicate the DSH technique for the disassembly-reassembly network models should give similar estimates with accuracy bounded by the values in Chapter 3. Therefore, a simple enhancenent to the DSH technique was implemented based upon characterization of the mean and variance of the maximum of two Erlang-type distributions, as derived by Equation 4.1.

For simplification purposes the distributions of the subroute sojourn times were approximated as special-Erlang with the number of stages n_i equal to the sum of the total number of stages of service occurring at each

shop when the particular subjob is at that shop. The estimated subroute sojourn time as computed by the DSH technique is divided by the number of stages, n_i, to give an approximate stage rate λ_i. Then, Equation 4.1 was solved for the two special-Erlang densities with n_1 and n_2 stages and stage rates, λ_1 and λ_2, respectively. This generalized form is given below

Let: $\mathcal{T} = \text{Max}\,[T_1, T_2]$

where $T_1 \tilde{}$ Erlang (λ_1, n_1) and $T_2 \tilde{}$ Erlang (λ_2, n_2)

then

$$\begin{aligned} g_{\mathcal{T}}(\tau) &= \frac{\lambda_1^{n_1}\tau^{n_1-1}}{(n_1-1)!}e^{\lambda_1\tau}\left(1 - e^{-\lambda_2\tau}\sum_{1\le i\le n_2}\frac{\lambda_2^{n_2-i}}{(n_2-i)!}\tau^{n_2-i}\right) \\ &+ \frac{\lambda_2^{n_2}\tau^{n_2-1}}{(n_2-1)!}e^{\lambda_2\tau}\left(1 - e^{-\lambda_1\tau}\sum_{1\le i\le n_2}\frac{\lambda_1^{n_1-i}}{(n_1-i)!}\tau^{n_1-i}\right) \end{aligned} \tag{4.2}$$

Given Equation 4.2 then the mean of $\mathcal{T}$ is specified by

$$\begin{aligned} E[\mathcal{T}] &= \int_0^\infty \tau g_{\mathcal{T}}(\tau)d\tau \\ &= \frac{n_1}{\lambda_1} + \frac{n_2}{\lambda_2} - \\ &\quad \left(\frac{\lambda_1^{n_1}}{(n_1-1)!}\sum_{i=1}^{n_2}\frac{\lambda_2^{n_2-i}(n_1+n_2-i)!}{(\lambda_1+\lambda_2+1-i)^{n_1+n_2+1-i}(n_2-1)!}\right) - \\ &\quad \left(\frac{\lambda_2^{n_2}}{(n_2-1)!}\sum_{i=1}^{n_1}\frac{\lambda_1^{n_1-i}(n_1+n_2-i)!}{(\lambda_1+\lambda_2+1-i)^{n_1+n_2+1-i}(n_1-1)!}\right) \end{aligned} \tag{4.3}$$

and the second moment is

$$
\begin{aligned}
E[\mathcal{T}^2] &= \int_0^\infty \tau^2 g_{\mathcal{T}}(\tau)\, d\tau \\
&= \frac{n_1(n_1+1)}{\lambda_1^2} + \frac{n_2(n_2+1)}{\lambda_2^2} - \\
&\quad \left(\frac{\lambda_1^{n_1}}{(n_1-1)!} \sum_{i=1}^{n_2} \frac{\lambda_2^{n_2-i}(n_1+n_2+2-i)!}{(\lambda_1+\lambda_2)^{n_1+n_2+1-i}(n_2-i)!} \right) - \\
&\quad \left(\frac{\lambda_2^{n_2}}{(n_2-1)!} \sum_{i=1}^{n_1} \frac{\lambda_1^{n_1-i}(n_1+n_2+1-i)!}{(\lambda_1+\lambda_2)^{n_1+n_2+2-i}(n_1-i)!} \right)
\end{aligned}
\tag{4.4}
$$

And, finally, the variance of $\mathcal{T}$ is given by

$$Var[\mathcal{T}] = E[\mathcal{T}^2] - E[\mathcal{T}]^2.$$

The sojourn time for a particular customer at the back of the queue of shop 1 can be broken into three portions, ie.,

$$\mathcal{T} = P_1 + P_2 + P_3$$

The first and third components, P_1 and P_3, are relatively simple to approximate. The first portion, P_1, is the time spent at shop 1 which can be deterministically approximated as the mean service time at shop 1 times the number of jobs at the shop upon arrival of the particular job (including the particular job). The third portion, P_3, is the time the mated particular subjobs spend at shop N waiting to be reassembled into the parent (particular) job. This can be deterministically approximated as the mean service time at shop N times the number of pairs of subjobs at

shop N upon arrival of the second of the particular subjobs (including the particular pair).

The second component of the sojourn time, P_2, is the maximum of times for each particular subjob to traverse its serial subroute. This is the portion that straight forward application of the DSH technique from Chapter 3 potentially could have trouble approximating because of the effects previously mentioned. An alternative is to use Equation 4.1 to account for the mean and variance associated with the maximum of the subroute sojourn times. However, given the analytic results of the previous chapter, the exact distributions associated with the serial subroutes are generalized hyper-Erlang. Thus, straightforward solution of Equation 4.1 in this case is not computationally feasible. However, as shown in Equations 4.2-4.4 this determination is relatively simple when the subroute sojourn times have special Erlang distributions.

An enhancement to the DSH technique, DSHSM, takes advantage of this. The DSHSM technique approximates portions P_1 and P_3 of the sojourn

time of the particular job deterministically as just described. The second portion, P_2, is approximated in the following fashion.

The DSH technique is used to determine the length of each of the two subroutes. The number of service completions at each shop during the particular subjob stay is computed deterministically for each subroute. The deterministic sojourn time for each subroute is then divided by the total of the number of individual shop completions for all shops in the subroute. This yields a weighted average service time. The total sojourn time for each subroute is then approximated as a special Erlang with the stage rate equal to the weighted average service time just described and the number of stages equal to the total number of shop completions.

Finally, the second portion of the total sojourn time, P_2, is approximated by determining the maximum of the two resulting special Erlang subroute sojourn approximations using Equation 4.2. Equation 4.3 and 4.4 are used to compute the mean and variance associated with the maximum.

See Appendix B.3 for the listing of the FORTRAN program.

4.3.4 Enhanced Deterministic Simulation Heuristic with Subroute Maximization (DPLSM)

This heuristic couples the approximation scheme for the maximum of particular subjob routes described in Section 4.3.3 with the DPL technique from Chapter 3.

The difference between this heuristic and the DSHSM just described is that the DPLSM technique utilizes the DPL heuristic to approximate the sojourn time associated with portions P_1 and P_3, as well as, the subroute sojourn estimates of time and number of completions for use in approximating portion P_2 through the use of Equations 4.2-4.4.

See Appendix B.4 for the listing of the FORTRAN program.

4.4 Experimental Design

An experimentation plan was developed in order to evaluate the effectiveness of the DSH, DPL, DSHSM, and DPLSM heuristics versus an empirical approximation based upon the Monte-Carlo simulation model.

As in the Chapter 3 scenarios, the plan consisted of a set of example scenarios which were designed to test the heuristics under conditions similar to those found in the NARF-like facilities described in Chapter 1. The

dominating factor of these facilities is heavy traffic. Scenarios which did not fall into this category were also included to test the robustness of the heuristics.

For the purposes of this dissertation symmetric networks were considered. That is, the serial subroutes of the two subjobs generated at shop 1 have identical initial queue distribution and service rates.

The experimentation was designed around four key parameters: the number of shops, the initial queue lengths, the shop service rates, and the service distributions. The number of shops was set at two levels; 4 and 8. The queue lengths were varied from 1 to 10. The service rates were normalized to a base value of 1.0. For purposes of emulating NARF-like facilities, rates between 0.8 and 1.0 were used.

A set of examples was developed to test a full design of the levels of each of these three factors. These examples are displayed in Table 4.4 and Table 4.5. Each example was solved for each of 5 different service time distributions. These distributions included the exponential distribution; three special Erlang distributions where the number of stages took on the

Table 4.4: Example Numbers for 4 Queue Scenarios

Example Number	Service Rate at Shops 1 - 4						
	1.0	1.0	1.0	0.8	1.0	0.8	0.8
	1.0	1.0	0.8	1.0	0.8	1.0	0.8
Queue Size at Shop	1.0	1.0	0.8	1.0	0.8	1.0	0.8
1, 2, 3, 4	1.0	0.8	1.0	1.0	0.8	0.8	1.0
1,1,1,1	1	29	57	85	113	141	169
5,1,1,1	2	30	58	86	114	142	170
1,5,5,1	3	31	59	87	115	143	171
1,1,1,5	4	32	60	88	116	144	172
5,5,5,1	5	33	61	89	117	145	173
5,1,1,5	6	34	62	90	118	146	174
1,5,5,5	7	35	63	91	119	147	175
5,5,5,5	8	36	64	92	120	148	176
8,1,1,1	9	37	65	93	121	149	177
1,8,8,1	10	38	66	94	122	150	178
1,1,1,8	11	39	67	95	123	151	179
8,8,8,1	12	40	68	96	124	152	180
8,1,1,8	13	41	69	97	125	153	181
1,8,8,8	14	42	70	98	126	154	182
8,5,5,5	15	43	71	99	127	155	183
5,8,8,5	16	44	72	100	128	156	184
5,5,5,8	17	45	73	101	129	157	185
8,8,8,5	18	46	74	102	130	158	186
8,5,5,8	19	47	75	103	131	159	187
5,8,8,8	20	48	76	104	132	160	188
8,8,8,8	21	49	77	105	133	161	189
7,7,7,7	22	50	78	106	134	162	190
7,8,8,8	23	51	79	107	135	163	191
8,7,7,8	24	52	80	108	136	164	192
8,8,8,7	25	53	81	109	137	165	193
7,7,7,8	26	54	82	110	138	166	194
7,8,8,7	27	55	83	111	139	167	195
8,7,7,7	28	56	84	112	140	168	196

values 2, 3, and 5; and a truncated normal distribution where the variance corresponded to an equivalent 10 stage Erlang. These distributions provided a means to determine the effect of the variance on the heuristics. The exponential distribution case provided the highest variance case and the variance was decreased in the 2 stage Erlang, and so on with the normal distribution providing the lowest variance case. Henceforth, the example datasets with the exponential service times are referred to as the Model 5 datasets, those with 2 stage Erlang distribution are Model 15, those with 3 stage are Model 25, those with 5 stage are Model 35, and finally those with the truncated normal are Model 45.

4.5 Results

In this section the results of the heuristic DSH, DPL, DSHSM, and DPLSM techniques are compared to those of the Monte-Carlo simulation model. These results are broken down by mean comparison and standard deviation comparison.

Table 4.5: Example Numbers for 8 Queue Scenarios

Example Number	Service Rate at Shops 1 - 8						
	1.0	0.8	1.0	1.0	0.8	0.9	0.85
	1.0	0.85	0.95	0.9	0.9	1.0	0.9
	1.0	0.85	0.95	0.9	0.9	1.0	0.9
	1.0	0.9	0.9	0.8	1.0	0.9	0.8
	1.0	0.9	0.9	0.8	1.0	0.9	0.8
	1.0	0.95	0.85	0.9	0.9	1.0	0.95
Queue Size at Shop	1.0	0.95	0.85	0.9	0.9	0.8	0.95
1, 2, 3, 4, 5, 6, 7, 8, 9, 10	1.0	1.0	0.8	1.0	0.8	0.9	1.0
10,10,10,10,10,10,10,10	197	213	229	245	261	277	293
10,8,8,9,9,10,10,7	198	214	230	246	262	278	294
7,8,8,9,9,10,10,10	199	215	231	247	263	279	295
10,9,9,8,8,9,9,10	200	216	232	248	264	280	296
8,9,9,10,10,9,9,8	201	217	233	249	265	281	297
7,10,10,5,5,8,8,9	202	218	234	250	266	282	298
6,10,10,9,9,7,7,8	203	219	235	251	267	283	299
10,9,9,10,10,8,8,10	204	220	236	252	268	284	300
8,8,8,8,8,10,10,10	205	221	237	253	269	285	301
10,10,10,6,6,9,9,8	206	222	238	254	270	286	302
9,10,10,9,9,8,8,7	207	223	239	255	271	287	303
10,10,10,9,9,9,9,9	208	224	240	256	272	288	304
9,7,7,8,8,9,9,10	209	225	241	257	273	289	305
5,5,5,5,5,5,5,5	210	226	242	258	274	290	306
5,6,6,7,7,7,7,6	211	227	243	259	275	291	307
9,9,9,10,10,10,10,9	212	228	244	260	276	292	308

4.5.1 Comparison of Means

In this section the approximated means of the total sojourn time for the example scenarios from each of the heuristics are compared with the empirical mean from the Monte-Carlo simulation. For purposes of comparison the percentage difference between the approximated mean sojourn time from the heuristics and the empirical solution was computed. The actual value of the sojourn means and standard deviations are of no importance since they are relative to the normalized service time distributions.

The percentage differences between the means from the heuristics and the simulation model are summarized by average absolute difference, average difference, and standard deviation of the average difference in Tables 4.6 - 4.8, respectively. Appendices D.1 - D.2 show the percentage for some selected models and queue sizes in the 4 and 8 queue scenarios.

Observing Table 4.6 note that the heuristics give good approximations as the average absolute difference is below 8% for the 4 queue scenarios for all techniques and below 5% for the 8 queue scenarios for all but the DPLSM heuristic in the Model 5 examples. For the 4 queue scenarios the average absolute difference of the DSH technique mean sojourn time

Table 4.6: Average Absolute Difference for the Mean

Model Number	Heuristic	Number of Queues 4	8
5	DSH	7.52	3.58
	DPL	5.53	4.39
	DSHSM	5.91	3.77
	DPLSM	7.36	8.37
15	DSH	4.82	1.60
	DPL	3.05	1.02
	DSHSM	4.74	3.36
	DPLSM	5.84	4.10
25	DSH	3.76	0.97
	DPL	2.57	0.73
	DSHSM	4.12	3.06
	DPLSM	5.53	3.30
35	DSH	2.81	0.51
	DPL	2.47	0.42
	DSHSM	3.47	2.59
	DPLSM	5.41	2.68
45	DSH	1.97	0.20
	DPL	2.71	0.19
	DSHSM	2.66	2.00
	DPLSM	5.18	2.07

Table 4.7: Average Difference for the Mean

Model Number	Heuristic	Number of Queues 4	8
5	DSH	7.38	3.49
	DPL	1.97	-1.47
	DSHSM	-0.05	-3.01
	DPLSM	-6.44	-8.23
15	DSH	4.49	1.48
	DPL	0.76	0.64
	DSHSM	-1.07	-3.24
	DPLSM	-5.46	-4.09
25	DSH	3.35	0.89
	DPL	-0.18	0.59
	DSHSM	-1.27	-3.00
	DPLSM	-5.32	-3.29
35	DSH	2.23	0.47
	DPL	-1.26	0.35
	DSHSM	-1.43	-2.56
	DPLSM	-5.33	-2.68
45	DSH	1.31	0.17
	DPL	-2.25	0.08
	DSHSM	-1.32	-1.98
	DPLSM	-5.17	-2.07

Table 4.8: Standard Deviation of the Average Difference for the Mean

Model Number	Heuristic	Number of Queues	
		4	8
5	DSH	8.15	3.50
	DPL	4.18	3.97
	DSHSM	4.77	2.09
	DPLSM	5.34	5.74
15	DSH	6.25	1.91
	DPL	3.35	1.47
	DSHSM	3.52	1.33
	DPLSM	4.66	1.40
25	DSH	5.28	1.34
	DPL	3.29	1.04
	DSHSM	3.07	0.89
	DPLSM	4.78	0.89
35	DSH	4.21	0.85
	DPL	3.78	0.64
	DSHSM	2.58	0.51
	DPLSM	5.23	0.54
45	DSH	3.12	0.49
	DPL	4.72	0.34
	DSHSM	2.11	0.29
	DPLSM	5.83	0.41

estimate ranged from 7.52% to 1.97%, that of the DPL ranged from 5.53% to 2.47%, that of the DSHSM ranged from 5.91% to 2.66%, and for the DPLSM it ranged from 7.36% to 5.18%. For the 8 queue scenarios the average absolute difference of the DSH technique mean sojourn time estimate ranged from 3.58% to 0.20%, that of the DPL ranged from 4.39% to 0.19%, that of the DSHSM ranged from 3.77% to 2.00%, and for the DPLSM it ranged from 8.37% to 2.07%.

These results are very much in line with those for the serial models covered in Chapter 3. Notice also the tendency for better results as the service variance decreases (from Model 5 down to Model 45). Comparisons between queue sizes should not be done because of the inconsistencies in the example scenarios (see Tables 4.6 and 4.5).

It is interesting to note that, in general, the enhanced heuristics which used Equations 4.2-4.4 (DSHSM and DPLSM) did not provide better approximations than their simpler counterparts (DSH and DPL). From Table 4.7 it can be seen that this is because the increase in the estimate provided by Equation 4.3 for the subroute portion of the mean sojourn time overshot the actual value (the average difference being consistently

negative). This overestimate is most probably due to using special Erlang approximations for the subroute sojourn which is actually generalized hyper-Erlang.

Overall the DPL technique appears to give the best approximations of the mean sojourn time with the DSH running a close second. However, all techniques seem to give very good approximations.

The estimators are all relatively consistent as evidenced by Table 4.8. The standard deviation of the absolute difference is generally less than 5%. These values are similar to those for the serial models of Chapter 3.

4.5.2 Comparison of Standard Deviations

The percentage differences between the standard deviation estimates from the heuristics and those from the simulation model for the 4 and 8 queue examples are summarized by average absolute difference, average difference, and standard deviation of the average difference in Tables 4.9 - 4.11, respectively. The actual differences for some selected models and queue sizes are given in Appendices D.3 - D.4.

The estimates for the standard deviation of the total sojourn time are good for all heuristics as evidenced by the average absolute differences

Table 4.9: Average Absolute Difference for the Standard Deviation

Model Number	Heuristic	Number of Queues 4	8
5	DSH	8.08	14.10
	DPL	9.51	13.96
	DSHSM	7.07	10.68
	DPLSM	7.77	11.99
15	DSH	8.22	12.45
	DPL	9.87	12.28
	DSHSM	6.79	10.23
	DPLSM	7.57	10.27
25	DSH	8.14	11.24
	DPL	9.75	11.07
	DSHSM	6.72	9.52
	DPLSM	7.69	9.53
35	DSH	8.18	9.09
	DPL	9.81	8.92
	DSHSM	6.89	8.27
	DPLSM	8.17	8.28
45	DSH	9.60	6.81
	DPL	11.20	6.65
	DSHSM	8.43	7.75
	DPLSM	11.00	7.77

Table 4.10: Average Difference for the Standard Deviation

Model Number	Heuristic	Number of Queues 4	8
5	DSH	-6.41	-13.24
	DPL	-8.26	-13.09
	DSHSM	-5.30	-6.16
	DPLSM	-6.16	-8.48
15	DSH	-6.59	-9.63
	DPL	-8.33	-9.47
	DSHSM	-4.14	-1.55
	DPLSM	-4.82	-1.64
25	DSH	-6.33	-8.05
	DPL	-8.01	-7.88
	DSHSM	-3.08	0.41
	DPLSM	-4.01	0.38
35	DSH	-5.83	-5.99
	DPL	-7.47	-5.82
	DSHSM	-2.00	2.84
	DPLSM	-3.19	2.81
45	DSH	-5.00	-3.29
	DPL	-6.71	-3.13
	DSHSM	-0.55	5.82
	DPLSM	-3.54	5.80

Table 4.11: Standard Deviation of the Average Difference for the Standard Deviation

Model Number	Heuristic	Number of Queues 4	8
5	DSH	5.93	11.42
	DPL	6.19	11.25
	DSHSM	5.49	6.85
	DPLSM	5.80	9.49
15	DSH	6.75	10.24
	DPL	7.31	9.98
	DSHSM	5.50	4.71
	DPLSM	5.49	4.78
25	DSH	7.23	9.29
	DPL	8.06	9.00
	DSHSM	5.39	3.94
	DPLSM	5.54	3.95
35	DSH	7.34	7.97
	DPL	8.44	7.70
	DSHSM	5.07	3.88
	DPLSM	5.69	3.89
45	DSH	8.57	6.04
	DPL	10.52	5.81
	DSHSM	6.68	4.44
	DPLSM	10.31	4.44

shown in Table 4.9. For the 4 queue scenarios the average absolute difference of the DSH technique sojourn time standard deviation estimate ranged from 9.60% to 8.08%, that of the DPL ranged from 11.20% to 9.51%, that of the DSHSM ranged from 8.43% to 6.72%, and for the DPLSM it ranged from 11.00% to 7.57%. For the 8 queue scenarios the average absolute difference of the DSH technique sojourn time standard deviation estimate ranged from 14.1% to 6.81%, that of the DPL ranged from 13.96% to 6.65%, that of the DSHSM ranged from 10.68% to 7.75%, and for the DPLSM it ranged from 11.99% to 7.77%.

As with the approximations for the mean sojourn time, the standard deviation estimates are very much in line with the approximations for the serial models of Chapter 3. However, the enhanced heuristics, DSHSM and DPLSM, appear to be slightly better than their counterparts, DSH and DPL. This is because the variance of the individual shop sojourn times are negatively correlated (as discussed in Chapter 3). From Equation 4.4 it can be seen that the approximations used in the DSHSM and DPLSM technique will be less than of the standard DSH and DPL heuristics. This is evidenced in Table 4.10 as the average difference is consistently less negative

for DSHSM and DPLSM than DSH and DPL. The DSHSM heuristic is generally the best estimator with the DPLSM a close second.

The consistency of the estimates can be discerned from Table 4.11. All the heuristics are relatively stable and on par with corresponding values in Chapter 3. Again the DSHSM heuristic appears to be the best with slightly more consistent estimates.

4.6 Conclusions

This chapter has been devoted to the study of transient sojourn times for jobs which require disassembly, service, and reassembly. In particular, the case was studied where two subjobs are generated at disassembly, each of which then traverses a serial network of single-server, FIFO queues like those described in Chapter 3. Upon completion of their respective subroutes the subjobs are reassembled into the parent job which then exits the system.

Exact solution of the sojourn time distribution for particular jobs under scenarios conditions is more complicated than that for the serial model case. Therefore, simple heuristic schemes were developed as an alternative to Monte-Carlo simulation. Each of the four heuristics developed were shown

to give good approximations of the mean and variance of the sojourn time for a range of example scenarios.

The best estimates of the mean sojourn time were provided by the DPL heuristic while the DSHSM technique gave the best estimates of the standard deviation. A hybrid of the two could be used to provide the best estimates for both parameters.

Two of the heuristics (DSHSM and DPLSM) incorporated a relatively simple determination of the maximum of two subroute sojourn times. For purposes of tractibility these subroute sojourn times were assumed to be special Erlang. The associated mean and variance of the maximum were used in an attempt to better capture the total sojourn time. While these heuristics did not provide significantly better (or worse) results than their simpler counterparts, they provide a basis for heuristic methodologies to solve more general disassembly-reassembly models exhibiting the generation of potentially many subjobs and non-symmetric subroutes.

CHAPTER 5

SUMMARY AND SUGGESTIONS FOR FUTURE RESEARCH

5.1 Summary

In this dissertation a study of heuristic methodologies for approximating sojourn times of particular jobs traversing networks of queues under "given scenario" conditions has been presented. Two basic kinds of networks were considered: serial networks of single-server, FIFO queues and disassembly-reassembly networks where jobs are disassembled into subjobs, serviced along different routes, and reassembled into the parent job. The study of these kinds of networks was motivated by real-world problems that exist in repair and maintenance facilities, such as Naval Air Rework Facilities (NARF's). Management problems that are encountered include the inability to accurately project completion times of particular jobs at a reasonable cost of time and resources.

A thorough search of the current literature revealed that this particular type of problem has been largely unstudied. Similar problems under

steady-state conditions have received a good deal of attention but the "given scenario" problems and the disassembly-reassembly job type have not been reported on to any extent.

An exact analytic procedure for solving the serial network problem was developed in Chapter 3. This procedure is an extension of the work of P.G. Harrison [30]. Harrison was primarily concerned with closed serial networks under exponential service. The procedure developed for this dissertation extended Harrison's procedure to more general Markovian service distributions and to open networks. From this procedure it was found that the exact sojourn distribution for the serial model is the so-called generalized hyper-Erlang, in other words, a convex combination of generalized Erlang distributions where each general Erlang term corresponds to a unique path through the associated Markovian state space. The shop sojourn time was also shown to be negatively correlated with particularly longer than average sojourn times at one shop coupled with shorter than average times at the next, and vice-versa.

Application of the exact technique to other than very small problems, however, was shown to be computationally infeasible. This is because of

the potentially huge numbers of potential paths through the state space of the model.

Given the computational difficulties of the exact technique, simple heuristics were developed as an alternative to time consuming Monte-Carlo simulation. Two heuristics based upon the Central Limit Theorem incorporated the idea of simple determinisitic simulation. Both techniques require only a minimal amount of computer memory and provide reliable approximations in only a fraction of the time required to perform Monte-Carlo simulation.

A wide range of example scenarios designed to test the performance of the heuristics on "NARF-like" problems were solved. Other scenarios designed to test the robustness of the techniques were also solved. The results of these tests showed that both heuristics performed very well in approximating the mean and standard deviation of the sojourn time.

The disassembly-reassembly problem was presented in Chapter 4. Again, computational complexity of the exact solution required that heuristic schemes be developed. To this end four techniques were presented. Two of the heuristics were the same ones for the serial models of Chapter 3. The other

two were based upon the first two and on a technique to compute the mean and variance of the maximum of two subroute sojourn distributions.

A set of example scenarios were also solved with all the heuristics and compared with the results from Monte-Carlo simulation. All four heuristics were shown to give good approximations of the sojourn mean and standard deviation.

In conclusion, deterministic simulation and simple variations of this scheme were shown to be effective in approximating sojourn times in the class of problems studied. The heuristics were particularly effective under heavy traffic scenarios.

5.2 Suggestions for Future Work

This research is only the start of work in a potentially fruitful arena of network queueing problems. There are many possible directions that can be taken to enhance and further develop solution methodologies for the complex sojourn time analysis beyond the work in this dissertation.

In the area of serial queueing models, more general service distributions need to be studied including generalized Erlang distributions. The

ability to model more complex service centers with multiple servers and more general service disciplines is also an important potential enhancement. Incorporations of multiple job types which require different service distributions must be included in order to study more generalized models of serial queueing systems. Another modification is the ability to model arrivals from potentially many sources to any shop in the network. In the same fashion departures from any shop must also be included. A simple diagram of such a model is shown in Figure 5.1.

Another important problem lies in characterizing the correlation in the sojourn times at successive shops in the network. This characterization could help improve the variance estimate from the simple heuristics.

For the disassembly-reassembly network models there is an even larger set of potential directions of research. Along with the enhancements of multiple job types, multiple servers, more general service distributions, and more general queueing models, incorporation of non-symmetric and more generalized subroutes for the subjobs generated are prime areas for further research. Also the generation of multiple offspring needs to be considered, as well as, multiple levels of disassembly-reassembly.

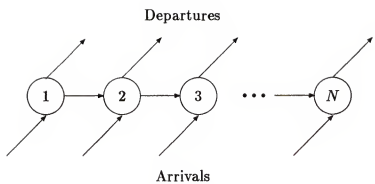


Figure 5.1: Generic Serial Queueing Network

APPENDICES

APPENDIX A

MISCELLANEOUS

A.1 Proof that the Distribution of the Minimum of M Exponentially Distributed Holding Times is Exponentially Distributed

Let

$$Y = Min[X_1, X_2, ...X_M] \tag{A.1}$$

where Y is a random variable and each X_i $(i = 1, ..., M)$ is an exponentially distributed random variable representing a holding time interval with rate μ_i. Then Y is also exponentially distributed with rate λ, where

$$\lambda = \sum_{i=1}^{M} \mu_i.$$

PROOF

The proof is by induction. First let $Y_1 = \text{Min}[X_1, X_2]$. By definition the density function associated with X_i is

$$f_{X_i}(t_i) = \mu_i e^{-\mu_i t_i}$$

then

$$
\begin{aligned}
F_{Y_1}(y_1) &= 1 - \int_{y_1}^{\infty} \int_{y_1}^{\infty} \mu_1 e^{-\mu_1 t_1} \mu_2 e^{-\mu_2 t_2} dt_1 dt_2 \\
&= 1 - \int_{y_1}^{\infty} \mu_1 e^{-\mu_1 t_1} dt_1 \int_{y_1}^{\infty} \mu_2 e^{-\mu_2 t_2} dt_2 \\
&= 1 - e^{-\mu_1 y_1} e^{-\mu_2 y_1} \\
&= 1 - e^{-(\mu_1 + \mu_2) y_1}
\end{aligned}
\tag{A.2}
$$

Equation A.2 is an exponential distribution, therefore Y_1 is an exponentially distributed random variable. Now, let $Y_2 = \text{Min}[Y_1, X_3]$. From Equation A.2, Y_2 is also exponentially distributed with rate $\mu_1 + \mu_2 + \mu_3$.

Iterating this process a total of $M - 1$ times leads to $Y = Y_{M-1}$ where Y is exponentially distributed with rate λ, where

$$\lambda = \sum_{i=1}^{M} \mu_i$$

and the proof is complete.

A.2 Derivation of T_i^* for the Small Example of Section 3.2.3

From Section 3.2.3 the Laplace-Stieljes transform of the sojourn time from state 1 to state 9 is shown to be

$$\begin{aligned} G_{1,9}^* &= T_1^* + T_2^* + T_3^* + T_4^* + T_5^* \\ &= \mu_1^2\mu_2^2\left(\frac{1}{(s+\mu_1+\mu_2)^2(s+\mu_2)^3} + \frac{1}{(s+\mu_1+\mu_2)^3(s+\mu_2)^3}\right) + \\ &\quad \frac{1}{(s+\mu_1)(s+\mu_1+\mu_2)^3(s+\mu_2)} + \frac{1}{(s+\mu_1)(s+\mu_1+\mu_2)^2(s+\mu_2)^2} + \\ &\quad \frac{1}{(s+\mu_1)^2(s+\mu_1+\mu_2)^2(s+\mu_2)}\Big) \end{aligned} \tag{A.3}$$

Now given the above, the problem is to reinvert Equation A.3 to the time domain. Let T_i^* represent term i in Equation A.3. Then

$$\begin{aligned} T_1^* &= \frac{1}{(s+\mu_1+\mu_2)^2(s+\mu_2)^3} \\ &= \frac{N_1(s)}{(s+\mu_1+\mu_2)^2} + \frac{N_2(s)}{(s+\mu_2)^3} \\ &\textit{where} \end{aligned} \tag{A.4}$$

$$N_1(s) = A_0 + A_1(s+\mu_1+\mu_2)$$

$$N_2(s) = B_0 + B_1(s+\mu_2) + B_2(s+\mu_2)^2$$

Next, Equation A.4 can be reinverted using partial fraction expansion to determine the unknown coefficients A_i and B_j. An example of the basic form of the reinversion is

$$\mathcal{L}^{-1}\left[\frac{N_1(s)}{(s+\mu_1+\mu_2)^2}\right] = e^{-(\mu_1+\mu_2)t}\sum_{j=0}^{1} A_{2-1-j}\frac{t^j}{j!}$$

Now, the unknown coefficients are determined as follows

Multiplying (A.4) by $(s+\mu_1+\mu_2)^2$ yields

$$\frac{1}{(s+\mu_2)^3} = A_0 + A_1(s+\mu_1+\mu_2) + \frac{N_2(s)(s+\mu_1+\mu_2)^2}{(s+\mu_2)^3}$$

To solve for A_0, let $s = -(\mu_1+\mu_2) \Rightarrow A_0 = \frac{-1}{\mu_1^3}$.

Similarly

$$\begin{aligned}
A_1 &= \frac{d}{ds}\left[\frac{1}{(s+\mu_2)^3}\right]_{s=-(\mu_1+\mu_2)} \\
&= \frac{-3}{(s+\mu_2)^4}\Big|_{s=-(\mu_1+\mu_2)} \\
&= \frac{-3}{\mu_1^4} \\
B_0 &= \frac{1}{(s+\mu_1+\mu_2)^2}\Big|_{s=-\mu_2} = \frac{1}{\mu_1^2} \\
B_1 &= \frac{-2}{(s+\mu_1+\mu_2)^3}\Big|_{s=-\mu_2} = \frac{-2}{\mu_1^3} \\
B_2 &= \frac{6}{(s+\mu_1+\mu_2)^4}\Big|_{s=-\mu_2} = \frac{6}{\mu_1^4}
\end{aligned}$$

Therefore

$$\begin{aligned}
T_1^* &= \left[\frac{-1}{\mu_1^3}\frac{1}{(s+\mu_1+\mu_2)^2} - \frac{3}{\mu_1^4}\frac{1}{(s+\mu_1+\mu_2)} + \frac{1}{\mu_1^2}\frac{1}{(s+\mu_2)^3} - \right. \\
&\quad \left. \frac{2}{\mu_1^3}\frac{1}{(s+\mu_2)^2} + \frac{6}{\mu_1^4}\frac{1}{(s+\mu_2)}\right]\mu_1\mu_2 \\
T_1(t) &= \mu_1^2\mu_2^2\left[\frac{-1}{\mu_1^3}te^{-(\mu_1+\mu_2)t} - \frac{3}{\mu_1^4}e^{-(\mu_1+\mu_2)t} + \right. \\
&\quad \left. \frac{1}{2\mu_1^2}t^2e^{-\mu_2^2t} - \frac{2}{\mu_1^3}te^{-\mu_2 t} + \frac{6}{\mu_1^4}e^{-\mu_2 t}\right]
\end{aligned} \tag{A.5}$$

Next term T_2^* is inverted in a similar fashion

$$T_2^* = \frac{1}{(s+\mu_1+\mu_2)^3(s+\mu_2)^2} = \frac{N_1(s)}{(s+\mu_1+\mu_2)^3} + \frac{N_2(s)}{(s+\mu_2)^2}$$

where

$$N_1(s) = A_0 + A_1(s+\mu_1+\mu_2) + A_2(s+\mu_1+\mu_2)^2$$

$$N_2(s) = B_0 + B_1(s+\mu_2)$$

and

$$A_0 = \frac{1}{(s+\mu_2)^2}\Big|_{s=-(\mu_1+\mu_2)} = \frac{1}{\mu_1^2}$$

$$A_1 = \frac{1}{(s+\mu_2)^3}\Big|_{s=-(\mu_1+\mu_2)} = \frac{2}{\mu_1^3}$$

$$A_2 = \frac{1}{(s+\mu_2)^4}\Big|_{s=-(\mu_1+\mu_2)} = \frac{6}{\mu_1^4}$$

$$B_0 = \frac{1}{(s+\mu_1+\mu_2)^3}\Big|_{s=-\mu_2} = \frac{1}{\mu_1^3}$$

$$B_1 = \frac{-3}{(s+\mu_1+\mu_2)^4}\Big|_{s=-\mu_2} = \frac{-3}{\mu_1^4}$$

$$T_2^* = \left[\frac{1}{\mu_1^2}\frac{1}{(s+\mu_1+\mu_2)^3} + \frac{2}{\mu_1^3}\frac{1}{(s+\mu_1+\mu_2)^2} + \right.$$
$$\left. \frac{6}{\mu_1^4}\frac{1}{(s+\mu_1+\mu_2)} + \frac{1}{\mu_1^3}\frac{1}{(s+\mu_2)^2} + \frac{-3}{\mu_1^4}\frac{1}{(s+\mu_2)}\right]$$

$$T_2(t) = \left[\frac{1}{2\mu_1^2}t^2e^{-(\mu_1+\mu_2)t} + \frac{2}{\mu_1^3}te^{-(\mu_1+\mu_2)t} + \right.$$
$$\left. \frac{6}{\mu_1}e^{-(\mu_1+\mu_2)t} + \frac{1}{\mu_1^3}te^{-\mu_2 t} - \frac{3}{\mu_1^4}e^{-\mu_2 t}\right]$$

Terms $T_3(t)$, $T_4(t)$, and $T_5(t)$ can be determined in the same fashion.

APPENDIX B

LISTINGS OF PROGRAMS FOR HEURISTICS

```
C*********************************************************************
C**                                                                 **
C**     Deterministic Simulation Heuristic (DSH)                    **
C**                                                                 **
C**     Developed By:   Russell E. King                             **
C**                     Dept. of Industrial and Systems Engineering **
C**                     University of Florida                       **
C**                     Gainesville, Florida                        **
C**                                                                 **
C**-----------------------------------------------------------------**
C**                                                                 **
C**     This routine computes the first two moments of sojourn      **
C**     distribution using the deterministic simulation heuristic,  **
C**     (DSH) for serial networks of single server queues.          **
C**                                                                 **
C*********************************************************************
      IMPLICIT INTEGER*4 (A-Z)
      REAL MU(10),VAR(10),SXX(10),SQSUM
      REAL STAGE(10),NUMC(10),DM1,DM2,DVAR,DSTD,STIME,NC
      DIMENSION QS(10),NEND(10),NS(10)
      CHARACTER CDATE*9,CTIME*8,MTYPE*15,STYPE*15
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      OPEN(UNIT=1,FILE='INPUT.DAT',STATUS='OLD')
      OPEN(UNIT=2,FILE='DSH.OUT',STATUS='NEW')
      CALL DATE(CDATE)
      CALL TIME(CTIME)
      READ(1,1) MTYPE,STYPE,EXAMPL,MODEL
1     FORMAT(////13X,A15,20X,A15/17X,I4,19X,I4)
      READ(1,10) NQ,NS
      READ(1,20) (QS(K),K=1,NQ)
10    FORMAT(//////46X,I5/46X,10I5)
20    FORMAT(///10(46X,I5//))
      READ(1,30) (MU(K),VAR(K),K=1,NQ)
30    FORMAT(///10(46X,F5.3,5X,F5.3/))
      NEND(1)=QS(1)
      DO 40 K=2,NQ
40    NEND(K)=NEND(K-1)+QS(K)
```

```
      WRITE(2,45) MTYPE,STYPE,EXAMPL,CDATE,MODEL,CTIME
45    FORMAT(25X,'DETERMINISTIC SIMULATION HEURISTIC RESULTS'/25X,
     $  42('-')///
     $  1X,'MODEL TYPE ==================> ',A15/
     $  1X,'REALM ======================> ',A15//
     $  1X,'*** NARF JOB SHOP EXAMPLE NUMBER',I3,2X,' ***',
     $  T45,'RUN DATE  ',A9//
     $  1X,'***    RESULTS FOR MODEL NUMBER',I4,2X,' ***',
     $  T45,'RUN TIME  ',A9///1X,80('*')/)
      WRITE(2,50) NQ,(K,QS(K),K=1,NQ)
50    FORMAT(///1X,'NUMBER OF QUEUES = ',I8,
     $  //1X,'QUEUE     SIZE'/
     $  1X,5('-'),5X,'----'/10(1X,I4,7X,I3/))
C-----------------------------------------------------------------------
C--                                                                   --
C--     Mark Beginning of Program Run Time                            --
C--                                                                   --
C-----------------------------------------------------------------------
      IF(.NOT.LIB$INIT_TIMER(INITIM)) GO TO 1000
C***********************************************************************
C**                                                                   **
C**     VARIABLE DEFINITIONS:                                         **
C**                                                                   **
C**     DM1         - FIRST MOMENT                                    **
C**                                                                   **
C**     DM2         - SECOND MOMENT                                   **
C**                                                                   **
C**     NUMC(K)     - NUMBER OF SERVICE COMPLETIONS AT STAGE K.       **
C**                                                                   **
C**     STAGE(K)    - SERVICE TIME REQUIRED AT STAGE K.               **
C**                                                                   **
C**     STIME       - CUMULATIVE SERVICE TIME                         **
C**                                                                   **
C***********************************************************************
      STAGE(1)=QS(1)/MU(1)
      NUMC(1)=QS(1)
      STIME=0
```

```
      DO 60 K=2,NQ
      STIME=STIME+STAGE(K-1)
      NC=MU(K)*STIME
      NUMC(K)=AMAX1(NEND(K)-NC,1.0)
      STAGE(K)=NUMC(K)/MU(K)
60    CONTINUE
      DM1=STIME+STAGE(NQ)
      DVAR=0.0
      DO 70 K=1,NQ
      SXX(K)=NUMC(K)*VAR(K)
      DVAR=DVAR+SXX(K)
70    CONTINUE
      DSTD=SQRT(DVAR)
      WRITE(2,80) DM1,DM2,DVAR,DSTD
80    FORMAT(//1X,80('*')//
     $       ' FIRST MOMENT=',F15.7,10X,'SECOND MOMENT=',F15.7//
     $       ' VARIANCE    =',F15.7,10X,'STANDARD DEV.=',F15.7)
      WRITE(2,90)
90    FORMAT(//1X,'EXPECTED NUMBER OF CUSTOMERS AT EACH STAGE'//
     $          1X,'STAGE',5X,'NUMBER')
      WRITE(2,100) (K,NUMC(K),K=1,NQ)
100   FORMAT(I4,7X,F5.2)
C-------------------------------------------------------------------------
C--                                                                     --
C--     Compute and Output Program Run Time                             --
C--                                                                     --
C-------------------------------------------------------------------------
      IF(.NOT.LIB$SHOW_TIMER(INITIM)) GO TO 1000
1000  STOP
      END
```

B.2 Listing of the DPL Program

```
C*********************************************************************
C**                                                                 **
C**     Enhanced Deterministic Simulation Heuristic (DPL)           **
C**                                                                 **
C**     Developed By:   Russell E. King                             **
C**                     Dept. of Industrial and Systems Engineering **
C**                     University of Florida                       **
C**                     Gainesville, Florida                        **
C**                                                                 **
C**-----------------------------------------------------------------**
C**                                                                 **
C**     This routine computes the first two moments of sojourn      **
C**     distribution using the enhanced deterministic simulation    **
C**     heuristic, (DPL) for serial networks of single server queues.**
C**                                                                 **
C*********************************************************************
      IMPLICIT INTEGER*4 (A-Z)
      REAL MU(10),ADD,PO,U1,U2,PSUM,PFACT,P1,PROB(15,50)
      REAL STAGE(10),NUMC(10),DM1,DM2,DVAR,DSTD,STIME,NC
      REAL NUMB,SXX(10),SQSUM
      DIMENSION NS(10),QS(10),NEND(10),POSSIB(15)
      CHARACTER CDATE*9,CTIME*8,MTYPE*15,STYPE*15
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      OPEN(UNIT=1,FILE='INPUT.DAT',STATUS='OLD')
      OPEN(UNIT=2,FILE='DPL.OUT',STATUS='NEW')
      CALL DATE(CDATE)
      CALL TIME(CTIME)
      READ(1,1) MTYPE,STYPE,EXAMPL,MODEL
1     FORMAT(////13X,A15,20X,A15/17X,I4,19X,I4)
      READ(1,10) NQ,(NS(I),I=1,NQ)
      READ(1,20) (QS(K),K=1,NQ)
10    FORMAT(//////46X,I5/46X,10I3)
20    FORMAT(///10(46X,I5/))
      READ(1,30) (MU(K),K=1,NQ)
30    FORMAT(///10(46X,F5.3/))
      NEND(1)=QS(1)
      DO 40 K=2,NQ
40    NEND(K)=NEND(K-1)+QS(K)
```

```
      WRITE(2,45) MTYPE,STYPE,EXAMPL,CDATE,MODEL,CTIME
45    FORMAT(25X,'DETERMINISTIC SIMULATION HEURISTIC RESULTS'/25X,
     $  42('-')///
     $  1X,'MODEL TYPE ==================> ',A15/
     $  1X,'REALM ========================> ',A15//
     $  1X,'*** NARF JOB SHOP EXAMPLE NUMBER',I3,2X,' ***',
     $  T45,'RUN DATE  ',A9//
     $  1X,'***    RESULTS FOR MODEL NUMBER',I4,2X,' ***',
     $  T45,'RUN TIME  ',A9///1X,80('*')/)
      WRITE(2,50) NQ,(K,QS(K),K=1,NQ)
50    FORMAT(///1X,'NUMBER OF QUEUES = ',I8//1X,'QUEUE     SIZE'/
     $  1X,5('-'),5X,'----'/10(1X,I4,7X,I3/))
C-------------------------------------------------------------------
C--                                                               --
C--     Mark Beginning of Program Run Time                        --
C--                                                               --
C-------------------------------------------------------------------
      IF(.NOT.LIB$INIT_TIMER(INITIM)) GO TO 1000
C*******************************************************************
C**                                                               **
C**     VARIABLE DEFINITIONS:                                     **
C**                                                               **
C**     DM1         - FIRST MOMENT                                **
C**                                                               **
C**     NUMC(K)     - NUMBER OF SERVICE COMPLETIONS AT STAGE K.   **
C**                                                               **
C**     STAGE(K)    - SERVICE TIME REQUIRED AT STAGE K.           **
C**                                                               **
C**     STIME       - CUMULATIVE SERVICE TIME                     **
C**                                                               **
C*******************************************************************
      STAGE(1)=QS(1)/MU(1)
      NUMC(1)=QS(1)
      STIME=0
```

```
      DO 60 K=2,NQ
      STIME=STIME+STAGE(K-1)
      NC=MU(K)*STIME
      NUMC(K)=AMAX1(NEND(K)-NC,1.0)
      STAGE(K)=NUMC(K)/MU(K)
60    CONTINUE
      DM1=STIME+STAGE(NQ)
C*********************************************************************
      NUMB=0.0
      DO 75 K=2,NQ
      U1=MU(K-1)
      U2=MU(K)
      NUMB=NUMB+NUMC(K-1)
      POSS=NEND(K)-U2/U1*(NUMB-1) + 0.99
      P0=U1/(U1+U2)
      IF(MU(K-1).LE.MU(K)) THEN
      IF(NUMC(K-1)/MU(K-1)-NUMC(K)/MU(K).GT.3.AND.POSS.LT.3) POSS=3
              IF(POSS.LE.1) THEN
                    POSS=2
              ENDIF
              PROB(K,POSS)=P0
              PFACT=(U2/(U1+U2))**NS(K)
              ADD=P0*POSS/MU(K)
              PSUM=P0
              DO 73 P=1,POSS-2
              P1=P0*PFACT**P
              PROB(K,POSS-P)=P1
              PSUM=PSUM+P1
              ADD=ADD+P1*(POSS-P)/MU(K)
73            CONTINUE
              POSSIB(K)=POSS
              PROB(K,1)=1.0-PSUM
              ADD=ADD+(1.0-PSUM)/MU(K)
              DM1=DM1-STAGE(K)
              DM1=DM1+AMAX1(ADD,STAGE(K))
```

```
      ELSE
            IF(POSS.LE.3) THEN
                  POSS=MAXO(2,POSS)
                  PROB(K,POSS)=PO
                  PFACT=(U2/(U1+U2))**NS(K)
                  ADD=PO*POSS/MU(K)
                  POSSIB(K)=POSS
                  PSUM=PO
                  DO 47 P=1,POSS-2
                  P1=PO*PFACT**P
                  PROB(K,POSS-P)=P1
                  PSUM=PSUM+P1
                  ADD=ADD+P1*(POSS-P)/MU(K)
47                CONTINUE
                  PROB(K,1)=1.0-PSUM
                  ADD=ADD+(1.0-PO)/MU(K)
                  DM1=DM1-STAGE(K)
                  DM1=DM1+AMAX1(ADD,STAGE(K))
            ELSE
                  POSSIB(K)=NUMC(K)
                  PROB(K,NUMC(K))=1.0
            ENDIF
       ENDIF
75     CONTINUE
C*********************************************************************
      SXX(1)=NUMC(1)/MU(1)**2/NS(1)
      DVAR=SXX(1)
      DO 170 K=2,NQ
      IF(POSSIB(K).GT.3) THEN
            SXX(K)=NUMC(K)/MU(K)**2/NS(K)
ELSE
            DO 70 P=1,POSSIB(K)
            SXX(K) = SXX(K) + PROB(K,P)*P/MU(K)**2/NS(K)
70          CONTINUE
      ENDIF
      DVAR=DVAR+SXX(K)
170   CONTINUE
      DSTD=SQRT(AMAX1(DVAR,0.0))
```

```
      WRITE(2,180) DM1,DM2,DVAR,DSTD
180   FORMAT(//1X,80('*')//
     $       ' FIRST MOMENT=',F15.7,10X,'SECOND MOMENT=',F15.7//
     $       ' VARIANCE    =',F15.7,10X,'STANDARD DEV.=',F15.7)
      WRITE(2,190)
190   FORMAT(//1X,'EXPECTED NUMBER OF CUSTOMERS AT EACH STAGE'//
     $          1X,'STAGE',5X,'NUMBER')
      WRITE(2,200) (K,NUMC(K),K=1,NQ)
200   FORMAT(I4,7X,F5.2)
C-----------------------------------------------------------------------
C--                                                                   --
C--     Compute and Output Program Run Time                           --
C--                                                                   --
C-----------------------------------------------------------------------
      IF(.NOT.LIB$SHOW_TIMER(INITIM)) GO TO 1000
1000  STOP
      END
```

B.3 Listing of the DSHSM Program

```
C**********************************************************************
C**                                                                  **
C**     Deterministic Simulation Heuristic with Subroute             **
C**     Maximization (DSHSM)                                         **
C**                                                                  **
C**     Developed By:   Russell E. King                              **
C**                     Dept. of Industrial and Systems Engineering  **
C**                     University of Florida                        **
C**                     Gainesville, Florida                         **
C**                                                                  **
C**------------------------------------------------------------------**
C**                                                                  **
C**     This routine computes the first two moments of sojourn       **
C**     distribution using the deterministic simulation heuristic    **
C**     with the subroute maximization enhancement (DSHSM)           **
C**     for disassembly-reassembly networks of single server queues. **
C**                                                                  **
C**********************************************************************
      IMPLICIT INTEGER*4 (A-Z)
      REAL*8 MU(10),SXX(10),CNTC,AM
      REAL*8 STAGE(10),NUMC(10),DM1,DM2,DVAR,DSTD,STIME,NC
      REAL*16 TERM1,TERM2,SUM1,SUM2,N2,FN,ETAU,ETAU2,
     $        AVGMU,VTAU
      DIMENSION NS(10),QS(10),NEND(10)
      CHARACTER CDATE*9,CTIME*8,MTYPE*15,STYPE*15
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      OPEN(UNIT=1,FILE='INPUT.DAT',STATUS='OLD')
      OPEN(UNIT=2,FILE='DSHSM.OUT',STATUS='NEW')
      CALL DATE(CDATE)
      CALL TIME(CTIME)
      READ(1,1) MTYPE,STYPE,EXAMPL,MODEL
1     FORMAT(////13X,A15,20X,A15/17X,I4,19X,I4)
      READ(1,10) NQ,(NS(I),I=1,NQ)
      READ(1,20) (QS(K),K=1,NQ)
10    FORMAT(//////46X,I5/46X,10I3)
20    FORMAT(///10(46X,I5/))
      READ(1,30) (MU(K),K=1,NQ)
30    FORMAT(///10(46X,F5.3/))
      NEND(1)=QS(1)
      DO 40 K=2,NQ
40    NEND(K)=NEND(K-1)+QS(K)
```

```
      WRITE(2,45) MTYPE,STYPE,EXAMPL,CDATE,MODEL,CTIME
45    FORMAT(25X,'DETERMINISTIC SIMULATION HEURISTIC RESULTS'/25X,
     $  42('-')///
     $  1X,'MODEL TYPE ==================> ',A15/
     $  1X,'REALM ======================> ',A15//
     $  1X,'*** NARF JOB SHOP EXAMPLE NUMBER',I3,2X,' ***',
     $  T45,' RUN DATE  ',A9//
     $  1X,'***    RESULTS FOR MODEL NUMBER',I4,2X,' ***',
     $  T45,' RUN TIME  ',A9///1X,80('*'))
WRITE(2,50) NQ,NS,(K,QS(K),K=1,NQ)
50    FORMAT(///1X,'NUMBER OF QUEUES = ',I8,10X,'NUMBER OF STAGES',
     $  ' OF SERVICE = ',10I3
     $ //1X,'QUEUE     SIZE'/
     $  1X,5('-'),5X,'----'/10(1X,I4,7X,I3/))
C-----------------------------------------------------------------------
C--                                                                   --
C--     Mark Beginning of Program Run Time                            --
C--                                                                   --
C-----------------------------------------------------------------------
      IF(.NOT.LIB$INIT_TIMER(INITIM)) GO TO 1000
C**********************************************************************
C**                                                                   **
C**     VARIABLE DEFINITIONS:                                         **
C**                                                                   **
C**     DM1         - FIRST MOMENT                                    **
C**                                                                   **
C**     NUMC(K)     - NUMBER OF SERVICE COMPLETIONS AT STAGE K.       **
C**                                                                   **
C**     STAGE(K)    - SERVICE TIME REQUIRED AT STAGE K.               **
C**                                                                   **
C**     STIME       - CUMULATIVE SERVICE TIME                         **
C**                                                                   **
C**********************************************************************
      STAGE(1)=QS(1)/MU(1)
      NUMC(1)=QS(1)
      CNTC=0
      STIME=0
```

```
      DO 60 K=2,NQ
      STIME=STIME+STAGE(K-1)
      NC=MU(K)*STIME
      NUMC(K)=DMAX1(NEND(K)-NC,1.0D0)
      STAGE(K)=NUMC(K)/MU(K)
      CNTC=CNTC+NUMC(K)*NS(K)
60    CONTINUE
      DM1=STIME+STAGE(NQ)
      AM=DM1-STAGE(1)-STAGE(NQ)
      CNTC=CNTC-NUMC(NQ)*NS(NQ)
      N=INT(CNTC+0.5)
      AVGMU=N/AM
      DM2=0.0
      DVAR=0.0
      DO 70 K=1,NQ
      SXX(K)=NUMC(K)/MU(K)**2/NS(K)
      DVAR=DVAR+SXX(K)
70    CONTINUE
      DSTD=SQRT(DVAR)
C*********************************************************************
C**                                                                 **
C**    Adjust mean and variance for maximum of two identical        **
C**    serial cases.  Note, however, that the first and last        **
C**    stages are not duplicated.  Therefore, there are three       **
C**    pieces to the total mean and variance.                       **
C**                                                                 **
C**    Piece 1 - mean and var of first queue sojourn time.          **
C**                                                                 **
C**    Piece 2 - maximum of two identical serial cases for queues   **
C**              2 - NQ-1.                                          **
C**                                                                 **
C**    Piece 3 - mean and var of last queue sojourn time.           **
C**                                                                 **
C**    Total mean = mean(1) + mean(2) + mean(3)                     **
C**    Total var  = var(1) + var(2) + var(3)  --- estimated         **
C**                                                                 **
C*********************************************************************
      TERM1=2.0
      NM1=N-1
      N2=N*2.0
      FN=N
```

```
      DO 75 I=1,NM1
 75   TERM1 = TERM1/4.0*(N2-I)/(FN-I)
      TERM1 = TERM1*FN/4.0
      TERM2 = TERM1*FN
      SUM1 = TERM1
      SUN2 = TERM2
      DO 78 I=1,NM1
      TERM2 = TERM1*(FN-I)
      TERM1 = TERM2*2.0/(N2-I)
      SUM1  = SUM1 + TERM1
 78   SUM2  = SUM2 + TERM2
      ETAU  = (N2-SUM1)/AVGMU
      ETAU2 = (N2*(FN+1)-SUM2)/AVGMU**2
      VTAU  = ETAU2 - ETAU**2
      DM1=STAGE(1)+STAGE(NQ)+ETAU
      DVAR=SXX(1)+SXX(NQ)+VTAU
      DSTD=SQRT(DVAR)
      WRITE(2,80) DM1,DM2,DVAR,DSTD
80    FORMAT(//80('*')//
     $       ,' FIRST MOMENT=',F15.7,10X,'SECOND MOMENT=',F15.7//
     $       ,' VARIANCE    =',F15.7,10X,'STANDARD DEV.=',F15.7)
      WRITE(2,90)
90    FORMAT(//1X,'EXPECTED NUMBER OF CUSTOMERS AT EACH STAGE'//
     $          1X,'STAGE',5X,'NUMBER')
      WRITE(2,100) (K,NUMC(K),K=1,NQ)
100   FORMAT(10X,I4,7X,F5.2)
C-------------------------------------------------------------------------
C--                                                                     --
C--     Compute and Output Program Run Time                             --
C--                                                                     --
C-------------------------------------------------------------------------
      IF(.NOT.LIB$SHOW_TIMER(INITIM)) GO TO 1000
1000  STOP
      END
```

B.4 Listing of the DPLSM Program

```
C*********************************************************************
C**                                                                 **
C**     Enhanced Deterministic Simulation Heuristic with Subroute   **
C**     Maximization (DPLSM)                                        **
C**                                                                 **
C**     Developed By:   Russell E. King                             **
C**                     Dept. of Industrial and Systems Engineering **
C**                     University of Florida                       **
C**                     Gainesville, Florida                        **
C**                                                                 **
C**-----------------------------------------------------------------**
C**                                                                 **
C**     This routine computes the first two moments of sojourn      **
C**     distribution using the enhanced deterministic simulation    **
C**     heuristic with the subroute maximization enhancement (DPLSM) **
C**     for disassembly-reassembly networks of single server queues. **
C**                                                                 **
C*********************************************************************
      IMPLICIT INTEGER*4 (A-Z)
      REAL*8 MU(10),PO,U1,U2,PSUM,PFACT,P1,PROB(15,50),NC,ADD
      REAL*8 STAGE(10),NUMC(10),DM1,DM2,DM3,DVAR,DSTD,STIME
      REAL*16 TERM1,TERM2,SUM1,SUM2,N2,FN,ETAU,ETAU2,
     $         AVGMU,VTAU,AM
      REAL NUMB,RHO,SXX(10),SQSUM,CNTC
      DIMENSION NS(10),QS(10),NEND(10),POSSIB(15)
      CHARACTER CDATE*9,CTIME*8,MTYPE*15,STYPE*15
CCCC
```

```
      WRITE(2,45) MTYPE,STYPE,EXAMPL,CDATE,MODEL,CTIME
45    FORMAT(25X,'DETERMINISTIC SIMULATION HEURISTIC RESULTS'/25X,
     $ 42('-')///
     $ 1X,'MODEL TYPE ==================> ',A15/
     $ 1X,'REALM ======================> ',A15//
     $ 1X,'*** NARF JOB SHOP EXAMPLE NUMBER',I3,2X,' ***',
     $ T45,'RUN DATE  ',A9//
     $ 1X,'***    RESULTS FOR MODEL NUMBER',I4,2X,' ***',
     $ T45,'RUN TIME  ',A9///1X,80('*')/)
      WRITE(2,50) NQ,(K,QS(K),K=1,NQ)
50    FORMAT(///1X,'NUMBER OF QUEUES = ',I8//1X,'QUEUE     SIZE'/
     $ 1X,5('-'),5X,'----'/10(1X,I4,7X,I3/))
C-------------------------------------------------------------------
C--                                                               --
C--     Mark Beginning of Program Run Time                        --
C--                                                               --
C-------------------------------------------------------------------
      IF(.NOT.LIB$INIT_TIMER(INITIM)) GO TO 1000
C*******************************************************************
C**                                                               **
C**     VARIABLE DEFINITIONS:                                     **
C**                                                               **
C**     DM1         - FIRST MOMENT                                **
C**                                                               **
C**     NUMC(K)     - NUMBER OF SERVICE COMPLETIONS AT STAGE K.   **
C**                                                               **
C**     STAGE(K)    - SERVICE TIME REQUIRED AT STAGE K.           **
C**                                                               **
C**     STIME       - CUMULATIVE SERVICE TIME                     **
C**                                                               **
C*******************************************************************
      STAGE(1)=QS(1)/MU(1)
      NUMC(1)=QS(1)
      CNTC=0.0
      STIME=0
```

```
      DO 60 K=2,NQ
      STIME=STIME+STAGE(K-1)
      NC=MU(K)*STIME
      NUMC(K)=DMAX1(NEND(K)-NC,1.0D0)
      STAGE(K)=NUMC(K)/MU(K)
      CNTC=CNTC+NUMC(K)*NS(K)
60    CONTINUE
      DM1=STIME+STAGE(NQ)
      AM=DM1-STAGE(1)-STAGE(NQ)
      CNTC=CNTC-NUMC(NQ)*NS(NQ)
      N=INT(CNTC+0.5)
C*********************************************************************
      NUMB=0.0
      DO 75 K=2,NQ
      U1=MU(K-1)
      U2=MU(K)
      UMB=NUMB+NUMC(K-1)
      POSS=NEND(K)-U2/U1*(NUMB-1) + 0.99
      PO=U1/(U1+U2)
      IF(MU(K-1).LE.MU(K)) THEN
      IF(NUMC(K-1)/MU(K-1)-NUMC(K)/MU(K).GT.3.AND.POSS.LT.3) POSS=3
            IF(POSS.LE.1) THEN
                  POSS=2
            ENDIF
            PROB(K,POSS)=PO
            PFACT=(U2/(U1+U2))**NS(K)
            ADD=PO*POSS/MU(K)
            PSUM=PO
            DO 73 P=1,POSS-2
            P1=PO*PFACT**P
            PROB(K,POSS-P)=P1
            PSUM=PSUM+P1
            ADD=ADD+P1*(POSS-P)/MU(K)
73          CONTINUE
            POSSIB(K)=POSS
            PROB(K,1)=1.0-PSUM
            ADD=ADD+(1.0-PSUM)/MU(K)
```

```
                IF(K.LT.NQ) THEN
                        AM=AM-STAGE(K)
                        AM=AM+DMAX1(ADD,STAGE(K))
                ELSE
                        DM3=DMAX1(ADD,STAGE(NQ))
                ENDIF
        ELSE
                IF(POSS.LE.3) THEN
                        POSS=MAXO(2,POSS)
                        PROB(K,POSS)=PO
                        PFACT=(U2/(U1+U2))**NS(K)
                        ADD=PO*POSS/MU(K)
                        POSSIB(K)=POSS
                        PSUM=PO
                        DO 47 P=1,POSS-2
                        P1=PO*PFACT**P
                        PROB(K,POSS-P)=P1
                        PSUM=PSUM+P1
                        ADD=ADD+P1*(POSS-P)/MU(K)
47                      CONTINUE
                        PROB(K,1)=1.0-PSUM
                        ADD=ADD+(1.0-PO)/MU(K)
                        IF(K.LT.NQ) THEN
                                AM=AM-STAGE(K)
                                AM=AM+DMAX1(ADD,STAGE(K))
                        ELSE
                                DM3=DMAX1(ADD,STAGE(NQ))
                        ENDIF
                ELSE
                        POSSIB(K)=NUMC(K)
                        PROB(K,NUMC(K))=1.0
                        IF(K.EQ.NQ) DM3=STAGE(NQ)
                ENDIF
        ENDIF
75      CONTINUE
```

```
C*****************************************************************
C**                                                             **
C**     Adjust mean and variance for maximum of two identical   **
C**     serial cases.  Note, however, that the first and last   **
C**     stages are not duplicated.  Therefore, there are three  **
C**     pieces to the total mean and variance.                  **
C**                                                             **
C**     Piece 1 - mean and var of first queue sojourn time.     **
C**                                                             **
C**     Piece 2 - maximum of two identical serial cases for queues **
C**               2 - NQ-1.                                     **
C**                                                             **
C**     Piece 3 - mean and var of last queue sojourn time.      **
C**                                                             **
C**     Total mean = mean(1) + mean(2) + mean(3)                **
C**     Total var  = var(1) + var(2) + var(3)  --- estimated    **
C**                                                             **
C*****************************************************************
      AVGMU=N/AM
      TERM1=2.0
      NM1=N-1
N2=N*2.0
      FN=N
      DO 175 I=1,NM1
175   TERM1 = TERM1/4.0*(N2-I)/(FN-I)
      TERM1 = TERM1*FN/4.0
      TERM2 = TERM1*FN
      SUM1 = TERM1
      SUM2 = TERM2
      DO 178 I=1,NM1
      TERM2 = TERM1*(FN-I)
      TERM1 = TERM2*2.0/(N2-I)
      SUM1  = SUM1 + TERM1
178   SUM2  = SUM2 + TERM2
      ETAU  = (N2-SUM1)/AVGMU
      ETAU2 = (N2*(FN+1)-SUM2)/AVGMU**2
      VTAU  = ETAU2 - ETAU**2
```

```
      DM1=STAGE(1)+DM3+ETAU
      SXX(1) = NUMC(1)/MU(1)**2/NS(1)
      DO 1790 K=2,NQ
      IF(POSSIB(K).GT.3) THEN
              SXX(NQ)= NUMC(K)/MU(K)**2/NS(K)
      ELSE
              SXX(NQ)=0.0
              DO 179 P=1,POSSIB(K)
              SXX(K) = SXX(K) + PROB(K,P)*P/MU(K)**2/NS(K)
179           CONTINUE
      ENDIF
1790  CONTINUE
      IF(N.LT.6) THEN
              VTAU=0.0
              DO 1792 K=2,NQ-1
1792          VTAU=VTAU+NUMC(K)/MU(K)**2/NS(K)
      ENDIF
      DVAR=SXX(1)+SXX(NQ)+VTAU
      DSTD=SQRT(DVAR)
      WRITE(2,80) DM1,DM2,DVAR,DSTD
80    FORMAT(//80('*')//
     $       ,' FIRST MOMENT=',F15.7,10X,'SECOND MOMENT=',F15.7//
     $       ,' VARIANCE    =',F15.7,10X,'STANDARD DEV.=',F15.7)
      WRITE(2,90)
90    FORMAT(//1X,'EXPECTED NUMBER OF CUSTOMERS AT EACH STAGE'//
     $          1X,'STAGE',5X,'NUMBER')
      WRITE(2,100) (K,NUMC(K),K=1,NQ)
100   FORMAT(10X,I4,7X,F5.2)
C-------------------------------------------------------------------------
C--                                                                     --
C--     Compute and Output Program Run Time                             --
C--                                                                     --
C-------------------------------------------------------------------------
      IF(.NOT.LIB$SHOW_TIMER(INITIM)) GO TO 1000
1000  STOP
      END
```

APPENDIX C

PERCENTAGE DIFFERENCES OF SELECTED EXAMPLES FOR SERIAL NETWORKS

C.1 Percentage Difference Between the MEANS of the EXACT Analytic Solution and the DSH and DPL Heuristics for the 2 Queue Model 1 Examples

EXMPL #	1	26	51	76	101	126	151
DSH	20.00	16.49	10.00	2.70	15.09	7.69	1.64
DPL	0.00	0.00	0.00	-4.95	0.00	0.00	0.00

EXMPL #	2	27	52	77	102	127	152
DSH	7.69	11.39	22.86	7.30	5.00	1.82	0.18
DPL	0.00	0.00	0.00	0.00	-7.82	-7.27	-4.62

EXMPL #	3	28	53	78	103	128	153
DSH	0.52	1.09	4.20	25.09	0.23	0.03	0.00
DPL	0.00	0.00	0.00	0.00	0.23	0.03	0.00

EXMPL #	4	29	54	79	104	129	154
DSH	0.04	0.13	0.86	11.44	0.01	0.00	0.00
DPL	0.00	0.00	0.00	0.00	0.01	0.00	0.00

EXMPL #	5	30	55	80	105	130	155
DSH	0.00	0.01	0.12	4.14	0.00	0.00	0.00
DPL	0.00	0.00	0.00	0.00	0.00	0.00	0.00

EXMPL #	6	31	56	81	106	131	156
DSH	22.58	17.68	9.40	1.94	16.70	7.95	1.43
DPL	9.68	7.23	3.36	-2.27	-16.76	-15.91	-9.89

EXMPL #	7	32	57	82	107	132	157
DSH	11.11	16.50	14.92	2.91	7.11	2.41	0.20
DPL	5.56	3.23	9.24	-1.25	7.11	2.41	0.20

EXMPL #	8	33	58	83	108	133	158
DSH	1.21	2.57	9.46	8.94	0.53	0.07	0.00
DPL	0.77	-1.83	4.35	5.03	0.53	0.07	0.00

EXMPL #	9	34	59	84	109	134	159
DSH	0.14	0.40	2.62	18.06	0.04	0.00	0.00
DPL	0.10	-2.20	1.48	7.54	0.04	0.00	0.00

EXMPL #	10	35	60	85	110	135	160
DSH	0.01	0.03	0.47	12.30	0.00	0.00	0.00
DPL	0.01	-1.76	0.31	5.01	0.00	0.00	0.00

EXMPL #	11	36	61	86	111	136	161
DSH	22.15	15.77	6.63	0.94	15.33	6.11	0.82
DPL	5.93	2.58	-0.92	-0.86	6.79	6.11	0.82

EXMPL #	12	37	62	87	112	137	162
DSH	14.20	18.90	7.68	0.98	8.60	2.42	0.14
DPL	11.13	6.19	0.22	-0.82	8.60	2.42	0.14

EXMPL #	13	38	63	88	113	138	163
DSH	3.14	6.74	14.30	1.30	1.29	0.14	0.00
DPL	2.84	5.63	7.38	-0.49	1.29	0.14	0.00

EXMPL #	14	39	64	89	114	139	164
DSH	0.61	1.81	10.38	2.20	0.18	0.01	0.00
DPL	0.58	1.65	7.66	0.43	0.18	0.01	0.00

EXMPL #	15	40	65	90	115	140	165
DSH	0.06	0.28	3.33	4.70	0.01	0.00	0.00
DPL	0.06	0.27	2.88	2.97	0.01	0.00	0.00

EXMPL #	16	41	66	91	116	141	166
DSH	20.62	13.75	4.95	0.61	13.48	4.69	0.55
DPL	9.60	4.84	-0.01	-0.54	13.48	4.69	0.55

EXMPL #	17	42	67	92	117	142	167
DSH	14.60	15.34	5.29	0.61	8.39	2.05	0.10
DPL	12.46	6.60	0.33	-0.54	8.39	2.05	0.10

EXMPL #	18	43	68	93	118	143	168
DSH	4.46	9.69	7.79	0.63	1.75	0.16	0.00
DPL	4.23	3.65	2.97	-0.51	1.75	0.16	0.00

EXMPL #	19	44	69	94	119	144	169
DSH	1.17	3.51	12.73	0.72	0.32	0.01	0.00
DPL	1.15	-1.13	8.17	-0.42	0.32	0.01	0.00

EXMPL #	20	45	70	95	120	145	170
DSH	0.17	0.78	8.03	1.07	0.03	0.00	0.00
DPL	0.17	-2.95	6.82	-0.07	0.03	0.00	0.00

EXMPL #	21	46	71	96	121	146	171
DSH	18.90	11.74	3.65	0.41	11.58	3.52	0.38
DPL	11.10	5.48	0.23	-0.36	11.58	3.52	0.38

EXMPL #	22	47	72	97	122	147	172
DSH	14.32	12.58	3.75	0.41	7.73	1.62	0.07
DPL	12.79	6.37	0.33	-0.36	7.73	1.62	0.07

EXMPL #	23	48	73	98	123	148	173
DSH	5.53	12.26	4.58	0.41	2.04	0.15	0.00
DPL	5.36	7.28	1.18	-0.36	2.04	0.15	0.00

EXMPL #	24	49	74	99	124	149	174
DSH	1.87	5.64	6.53	0.41	0.48	0.01	0.00
DPL	1.85	1.93	3.21	-0.36	0.48	0.01	0.00

EXMPL #	25	50	75	100	125	150	175
DSH	0.38	1.72	11.19	0.44	0.06	0.00	0.00
DPL	0.38	-1.36	8.03	-0.33	0.06	0.00	0.00

C.2 Percentage Difference Between the MEANS of the MONTE CARLO Simulation and the DSH and DPL Heuristics for the 10 Queue Model 11 Examples

EXMPL #	484	500	516	532	548	564	580
DSH	1.74	0.42	1.63	4.53	0.23	1.70	2.02
DPL	1.74	-0.45	1.00	-0.54	-2.26	-1.63	-1.14

EXMPL #	485	501	517	533	549	565	581
DSH	1.97	0.78	1.96	4.74	0.15	2.00	2.12
DPL	1.97	0.62	1.30	-0.24	-2.60	-1.43	-1.34

EXMPL #	486	502	518	534	550	566	582
DSH	3.22	1.27	2.75	6.95	0.56	1.90	2.29
DPL	3.22	-0.02	1.78	1.93	-2.85	-2.06	-1.61

EXMPL #	487	503	519	535	551	567	583
DSH	1.63	0.74	1.48	3.83	0.07	2.09	1.91
DPL	1.63	0.74	0.81	-0.73	-1.79	-1.27	-0.89

EXMPL #	488	504	520	536	552	568	584
DSH	2.59	1.13	2.38	6.64	0.47	1.96	2.14
DPL	2.59	-0.36	1.39	1.26	-3.30	-2.28	-2.07

EXMPL #	489	505	521	537	553	569	585
DSH	1.46	0.29	1.35	3.15	0.14	2.81	2.91
DPL	1.46	0.29	1.08	-1.11	-1.50	-0.62	-0.43

EXMPL #	490	506	522	538	554	570	586
DSH	4.35	1.76	4.14	8.06	1.10	2.17	2.18
DPL	4.35	1.76	2.66	2.56	-0.04	-2.72	-6.81

EXMPL #	491	507	523	539	555	571	587
DSH	2.68	0.63	2.47	5.62	0.43	1.93	2.00
DPL	2.68	0.55	1.71	0.27	-2.25	-2.26	-2.43

EXMPL #	492	508	524	540	556	572	588
DSH	2.39	0.78	2.20	5.52	0.53	2.39	2.52
DPL	2.39	0.78	1.16	1.26	-1.89	-2.70	-5.44

EXMPL #	493	509	525	541	557	573	589
DSH	3.31	1.22	2.97	7.60	0.76	2.51	2.72
DPL	3.31	-0.84	2.97	0.61	-1.67	-1.10	-1.90

EXMPL #	494	510	526	542	558	574	590
DSH	4.14	2.06	3.58	8.30	0.88	1.96	2.41
DPL	4.14	0.87	3.58	0.47	-1.67	-2.22	-1.41

EXMPL #	495	511	527	543	559	575	591
DSH	1.54	0.59	1.41	4.89	0.14	3.94	0.00
DPL	1.54	-2.47	1.11	-1.10	-4.56	-1.90	-0.62

EXMPL #	496	512	528	544	560	576	592
DSH	4.07	1.52	3.47	7.77	0.97	2.15	2.36
DPL	4.07	0.31	1.42	1.99	-3.31	-2.01	-2.48

EXMPL #	497	513	529	545	561	577	593
DSH	1.89	0.75	2.00	4.71	0.32	2.08	2.10
DPL	1.89	-0.02	0.65	-0.47	-1.87	-1.19	-0.80

EXMPL #	498	514	530	546	562	578	594
DSH	1.88	0.53	1.55	4.01	0.21	1.73	2.08
DPL	1.88	0.25	0.32	0.66	-1.87	-1.22	-1.30

EXMPL #	499	515	531	547	563	579	595
DSH	2.49	0.76	2.30	4.87	0.43	2.03	2.36
DPL	2.49	0.76	2.30	-0.82	-1.94	-1.05	-0.04

C.3 Percentage Difference Between the MEANS of the MONTE CARLO Simulation and the DSH and DPL Heuristics for the 5 Queue Model 41 Examples

EXMPL #	372	388	404	420	436	452	468
DSH	-0.04	0.06	0.02	0.28	0.09	-0.02	-0.03
DPL	-0.04	0.06	0.02	0.28	0.09	-0.02	-0.03

EXMPL #	373	389	405	421	437	453	469
DSH	0.06	0.35	-0.08	1.30	0.14	-0.04	0.39
DPL	0.06	-0.42	-0.08	1.28	0.14	-0.04	-0.38

EXMPL #	374	390	406	422	438	454	470
DSH	0.04	-0.06	-0.08	0.07	-0.02	0.04	0.18
DPL	0.04	-0.06	-0.08	0.07	-0.02	0.04	0.18

EXMPL #	375	391	407	423	439	455	471
DSH	-0.01	0.17	-0.01	0.13	0.07	-0.04	0.04
DPL	-0.01	0.17	-0.01	0.13	0.07	-0.04	0.04

EXMPL #	376	392	408	424	440	456	472
DSH	0.15	0.19	0.01	0.70	0.02	0.00	0.26
DPL	0.15	0.19	0.01	0.70	0.02	0.00	0.26

EXMPL #	377	393	409	425	441	457	473
DSH	-0.01	0.04	-0.02	0.24	-0.03	0.08	-0.04
DPL	-0.01	0.04	-0.02	0.24	-0.57	0.08	-0.04

EXMPL #	378	394	410	426	442	458	474
DSH	-0.14	0.01	-0.03	0.46	-0.11	0.00	0.06
DPL	-0.14	0.01	-0.03	0.46	-0.11	0.00	0.06

EXMPL #	379	395	411	427	443	459	475
DSH	-0.06	-0.03	0.06	0.11	0.00	-0.01	0.09
DPL	-0.06	-0.03	0.06	0.11	0.00	-0.01	0.09

EXMPL #	380	396	412	428	444	460	476
DSH	0.04	0.00	-0.04	0.14	-0.06	-0.11	-0.02
DPL	0.04	0.00	-0.04	0.14	-0.06	-0.11	-0.02

EXMPL #	381	397	413	429	445	461	477
DSH	0.02	0.23	0.03	0.50	-0.15	0.06	0.09
DPL	0.02	0.23	0.03	0.50	-0.27	0.06	0.09

EXMPL #	382	398	414	430	446	462	478
DSH	0.07	0.26	0.15	1.21	-0.06	0.08	0.33
DPL	0.07	-0.40	0.15	1.21	-0.06	0.08	-1.46

EXMPL #	383	399	415	431	447	463	479
DSH	0.03	0.03	0.04	0.46	0.03	-0.03	0.10
DPL	0.03	0.03	0.04	0.46	0.03	-0.03	0.10

EXMPL #	384	400	416	432	448	464	480
DSH	-0.04	0.12	-0.02	0.20	0.11	-0.10	-0.06
DPL	-0.04	0.12	-0.02	0.20	0.11	-0.10	-0.06

EXMPL #	385	401	417	433	449	465	481
DSH	0.10	0.46	0.14	2.53	0.20	0.10	0.59
DPL	0.10	0.46	0.14	-0.81	0.20	0.10	-1.26

EXMPL #	386	402	418	434	450	466	482
DSH	0.02	0.17	-0.02	1.07	-0.08	0.04	0.21
DPL	0.02	0.17	-0.02	1.07	-0.08	0.04	-0.49

EXMPL #	387	403	419	435	451	467	483
DSH	0.01	0.03	0.03	0.55	0.01	-0.11	0.06
DPL	0.01	0.03	0.03	0.55	0.01	-0.11	0.06

C.4 Percentage Difference Between the STANDARD DEVIATION of the EXACT Analytic Solution and the DSH and DPL Heuristics for the 2 Queue Model 1 Examples

EXMPL #	1	26	51	76	101	126	151
DSH	5.72	1.41	-3.51	-2.15	5.20	4.45	3.03
DPL	-5.41	-6.79	-6.90	-3.07	-3.55	0.00	2.15
EXMPL #	2	27	52	77	102	127	152
DSH	-1.06	-2.88	-4.48	-5.52	0.64	2.69	2.44
DPL	-5.19	-8.70	-13.39	-6.47	-2.09	1.71	2.34
EXMPL #	3	28	53	78	103	128	153
DSH	-0.80	-3.73	-14.46	-17.89	0.89	2.01	1.34
DPL	-0.80	-3.73	-14.46	-17.89	0.89	2.01	1.34
EXMPL #	4	29	54	79	104	129	154
DSH	-0.13	-2.02	-11.24	-33.90	0.84	1.39	0.89
DPL	-0.13	-2.02	-11.24	-33.90	0.84	1.39	0.89
EXMPL #	5	30	55	80	105	130	155
DSH	-0.01	-1.23	-7.67	-43.39	0.62	0.97	0.62
DPL	-0.01	-1.23	-7.67	-43.39	0.62	0.97	0.62
EXMPL #	6	31	56	81	106	131	156
DSH	5.50	-0.19	-4.68	-1.77	5.33	5.56	4.36
DPL	-2.07	-5.45	-6.60	-2.24	-10.30	-4.40	0.76

EXMPL #	7	32	57	82	107	132	157
DSH	-1.60	-3.31	-7.96	-3.14	0.71	4.07	3.76
DPL	-4.73	-10.18	-9.94	-3.62	6.26	4.07	11.95

EXMPL #	8	33	58	83	108	133	158
DSH	-1.83	-6.80	-19.27	-10.76	1.25	3.43	2.31
DPL	-1.83	-6.80	-19.27	-11.27	1.25	3.43	2.31

EXMPL #	9	34	59	84	109	134	159
DSH	-0.40	-3.97	-19.31	-21.63	1.46	2.52	1.61
DPL	-0.40	-3.97	-19.31	-21.63	1.46	2.52	1.61

EXMPL #	10	35	60	85	110	135	160
DSH	-0.04	-2.35	-14.25	-39.65	1.14	1.80	1.15
DPL	-0.04	-2.35	-14.25	-39.65	1.14	1.80	1.15

EXMPL #	11	36	61	86	111	136	161
DSH	3.02	-3.63	-5.26	-0.98	3.74	6.97	6.14
DPL	-6.61	-10.10	-7.47	-1.17	0.48	12.00	6.14

EXMPL #	12	37	62	87	112	137	162
DSH	-2.71	-4.59	-7.22	-1.09	0.32	6.24	5.63
DPL	-4.53	-11.12	-9.47	-1.28	0.32	6.24	5.63

EXMPL #	13	38	63	88	113	138	163
DSH	-4.63	-12.02	-15.55	-2.09	1.33	6.00	4.08
DPL	-4.63	-12.02	-17.97	-2.28	1.33	6.00	4.08

EXMPL #	14	39	64	89	114	139	164
DSH	-1.75	-9.31	-25.28	-4.50	2.48	4.89	3.13
DPL	-1.75	-9.31	-25.28	-4.69	2.48	4.89	3.13

EXMPL #	15	40	65	90	115	140	165
DSH	-0.30	-5.52	-27.99	-10.48	2.31	3.75	2.38
DPL	-0.30	-5.52	-27.99	-10.69	2.31	3.75	2.38

EXMPL #	16	41	66	91	116	141	166
DSH	1.06	-5.50	-4.83	-0.64	2.51	7.93	6.86
DPL	-5.59	-9.85	-6.23	-0.76	6.38	7.93	6.86

EXMPL #	17	42	67	92	117	142	167
DSH	-3.58	-6.85	-5.81	-0.65	-0.07	7.51	6.43
DPL	-4.87	-11.26	-7.23	-0.77	3.44	7.51	6.43

EXMPL #	18	43	68	93	118	143	168
DSH	-6.53	-13.49	-11.61	-0.78	1.09	7.45	5.05
DPL	-6.53	-13.49	-13.10	-0.90	1.09	7.45	5.05

EXMPL #	19	44	69	94	119	144	169
DSH	-3.31	-13.28	-19.33	-1.19	2.86	6.39	4.08
DPL	-3.31	-13.28	-20.92	-1.31	2.86	6.39	4.08

EXMPL #	20	45	70	95	120	145	170
DSH	-0.82	-8.68	-29.98	-2.57	3.07	5.13	3.25
DPL	-0.82	-8.68	-29.98	-2.69	3.07	5.13	3.25

EXMPL #	21	46	71	96	121	146	171
DSH	-0.81	-6.90	-4.09	-0.43	1.50	8.93	7.31
DPL	-5.54	-9.93	-5.03	-0.51	3.30	8.93	7.31

EXMPL #	22	47	72	97	122	147	172
DSH	-4.51	-8.17	-4.52	-0.43	-0.40	8.67	6.98
DPL	-5.44	-11.24	-5.46	-0.51	-0.40	8.67	6.98

EXMPL #	23	48	73	98	123	148	173
DSH	-8.12	-13.18	-7.51	-0.44	0.80	8.67	5.81
DPL	-8.12	-13.18	-8.48	-0.52	0.80	8.67	5.81

EXMPL #	24	49	74	99	124	149	174
DSH	-5.22	-16.26	-13.09	-0.47	3.01	7.72	4.92
DPL	-5.22	-16.26	-14.11	-0.55	3.01	7.72	4.92

EXMPL #	25	50	75	100	125	150	175
DSH	-1.76	-12.60	-22.32	-0.63	3.71	6.45	4.08
DPL	-1.76	-12.60	-23.42	-0.71	3.71	6.45	4.08

C.5 Percentage Difference Between the STANDARD DEVIATIONS of the MONTE CARLO Simulation and the DSH and DPL Heuristics for the 10 Queue Model 11 Examples

EXMPL #	484	500	516	532	548	564	580
DSH	-17.11	-6.17	-7.62	-38.31	-0.38	-4.32	-7.68
DPL	-17.11	-6.04	-2.93	-34.77	-0.20	-5.12	-8.48

EXMPL #	485	501	517	533	549	565	581
DSH	-14.80	-6.98	-9.95	-33.87	-1.40	-6.09	-8.40
DPL	-14.80	-7.06	-5.37	-31.59	-1.78	-6.98	-9.28

EXMPL #	486	502	518	534	550	566	582
DSH	-20.71	-10.87	-13.03	-39.65	-3.01	-4.20	-9.51
DPL	-20.71	-11.54	-9.55	-31.63	-3.49	-5.10	-10.42

EXMPL #	487	503	519	535	551	567	583
DSH	-14.18	-2.25	-11.75	-31.57	-0.58	-5.66	-7.05
DPL	-14.18	-2.23	-11.75	-31.57	0.15	-6.56	-7.93

EXMPL #	488	504	520	536	552	568	584
DSH	-19.16	-11.25	-9.94	-40.46	-3.22	-5.26	-9.82
DPL	-19.16	-12.02	-5.94	-38.19	-3.74	-6.15	-10.72

EXMPL #	489	505	521	537	553	569	585
DSH	-11.48	-4.31	-7.54	-26.72	0.62	-8.64	-11.90
DPL	-11.48	-3.33	-7.54	-26.72	1.83	-9.77	-13.05

EXMPL #	490	506	522	538	554	570	586
DSH	-19.14	-10.77	-12.26	-33.51	-6.63	-4.39	-10.00
DPL	-19.14	-9.98	-9.91	-30.28	-5.78	-5.93	-11.50

EXMPL #	491	507	523	539	555	571	587
DSH	-17.04	-7.39	-12.67	-35.30	-2.09	-4.94	-8.81
DPL	-17.04	-6.47	-8.62	-31.09	-1.53	-5.81	-9.69

EXMPL #	492	508	524	540	556	572	588
DSH	-16.56	-6.33	-10.01	-36.73	-1.83	-7.12	-9.56
DPL	-16.56	-5.85	-5.44	-32.20	-1.81	-8.16	-11.05

EXMPL #	493	509	525	541	557	573	589
DSH	-19.45	-10.81	-9.02	-36.82	-4.24	-6.22	-11.09
DPL	-19.45	-11.87	-5.01	-36.09	-5.38	-7.89	-12.20

EXMPL #	494	510	526	542	558	574	590
DSH	-21.58	-14.40	-10.31	-35.56	-4.56	-4.13	-7.41
DPL	-21.58	-14.23	-7.35	-31.67	-4.19	-5.10	-8.39

EXMPL #	495	511	527	543	559	575	591
DSH	-14.55	-6.51	-8.15	-39.25	-0.15	-14.88	4.65
DPL	-14.55	-8.01	-8.15	-38.46	-0.86	-16.36	4.61

EXMPL #	496	512	528	544	560	576	592
DSH	-20.68	-12.11	-14.43	-36.36	-3.32	-3.50	-9.36
DPL	-20.68	-11.47	-11.36	-30.83	-2.88	-4.47	-10.36

EXMPL #	497	513	529	545	561	577	593
DSH	-17.80	-6.19	-10.05	-34.24	-1.45	-4.26	-11.39
DPL	-17.80	-5.85	-5.25	-31.67	-1.67	-5.19	-12.34

EXMPL #	498	514	530	546	562	578	594
DSH	-14.70	-4.79	-9.64	-33.55	-0.34	-5.58	-8.66
DPL	-14.70	-3.81	-9.64	-30.50	-0.32	-6.45	-9.52

EXMPL #	499	515	531	547	563	579	595
DSH	-15.24	-6.30	-9.36	-32.32	-0.90	-4.42	-7.70
DPL	-15.24	-5.86	-5.01	-32.32	0.30	-5.27	-8.56

C.6 Percentage Difference Between the STANDARD DEVIATIONS of the MONTE CARLO Simulation and the DSH and DPL Heuristics for the 5 Queue Model 41 Examples

EXMPL #	372	388	404	420	436	452	468
DSH	1.74	-7.23	5.28	-11.15	3.97	2.27	-8.75
DPL	1.74	-7.23	5.28	-11.15	3.97	2.27	-8.75

EXMPL #	373	389	405	421	437	453	469
DSH	-1.38	-12.80	5.73	-20.70	5.76	2.96	-13.40
DPL	-1.38	-12.80	5.73	-20.70	5.76	2.96	-13.40

EXMPL #	374	390	406	422	438	454	470
DSH	1.84	-5.16	2.85	-9.46	4.94	3.02	-7.40
DPL	1.84	-5.16	2.85	-9.46	4.94	3.02	-7.40

EXMPL #	375	391	407	423	439	455	471
DSH	0.85	-4.34	6.65	-10.47	3.83	2.51	-5.71
DPL	0.85	-4.34	6.65	-10.47	3.83	2.51	-5.71

EXMPL #	376	392	408	424	440	456	472
DSH	-1.34	-9.34	6.69	-17.82	5.56	2.85	-9.69
DPL	-1.34	-9.34	6.69	-17.82	5.56	2.85	-9.69

EXMPL #	377	393	409	425	441	457	473
DSH	1.70	-7.42	3.42	-8.29	4.76	3.12	-5.98
DPL	1.70	-7.42	3.42	-8.29	4.76	3.12	-5.98

EXMPL #	378	394	410	426	442	458	474
DSH	1.22	-7.82	5.74	-16.09	4.55	2.46	-9.47
DPL	1.22	-7.82	5.74	-13.06	4.55	2.46	-8.95

EXMPL #	379	395	411	427	443	459	475
DSH	-1.21	-5.78	5.14	-9.50	3.72	2.12	-7.91
DPL	-1.21	-5.78	5.14	-6.42	3.72	2.12	-7.20

EXMPL #	380	396	412	428	444	460	476
DSH	1.84	-3.75	6.82	-7.52	5.18	2.96	-6.05
DPL	1.84	-3.75	6.82	-7.52	5.18	2.96	-6.05

EXMPL #	381	397	413	429	445	461	477
DSH	0.59	-10.58	7.74	-13.87	2.54	1.62	-9.41
DPL	0.59	-10.58	7.74	-13.87	2.54	1.62	-9.41

EXMPL #	382	398	414	430	446	462	478
DSH	0.19	-13.59	5.40	-23.88	4.47	2.06	-11.87
DPL	0.19	-13.59	5.40	-19.87	4.47	2.06	-10.83

EXMPL #	383	399	415	431	447	463	479
DSH	2.05	-9.00	5.31	-13.72	5.42	0.49	-7.82
DPL	2.05	-9.00	5.31	-9.16	5.42	0.49	-6.00

EXMPL #	384	400	416	432	448	464	480
DSH	1.63	-3.59	7.13	-6.22	3.33	2.93	-5.26
DPL	1.63	-3.59	7.13	-6.22	3.33	2.93	-5.26

EXMPL #	385	401	417	433	449	465	481
DSH	-1.47	-15.17	5.72	-17.67	4.45	1.30	-17.53
DPL	-1.47	-15.17	5.72	-19.38	4.45	1.30	-16.91

EXMPL #	386	402	418	434	450	466	482
DSH	-1.72	-11.60	6.26	-17.32	4.90	1.00	-11.75
DPL	-1.72	-11.60	6.26	-17.32	4.90	1.00	-11.75

EXMPL #	387	403	419	435	451	467	483
DSH	0.54	-8.94	6.88	-13.36	5.48	2.40	-7.25
DPL	0.54	-8.94	6.88	-13.36	5.48	2.40	-7.25

APPENDIX D

PERCENTAGE DIFFERENCES OF SELECTED EXAMPLES FOR DISASSEMBLY-REASSEMBLY

D.1 Percentage Difference Between the MEANS of the MONTE CARLO Simulation and the DSH, DPL, DSHSM and DPLSM Heuristics for the 4 Queue Model 25 Examples

EXMPL #	1	29	57	85	113	141	169
DSH	22.13	12.26	18.83	18.49	20.10	15.15	21.60
DPL	-3.83	-23.09	-1.79	-5.20	-2.69	-23.43	-2.36
DSHSM	14.02	4.94	9.32	10.65	11.36	8.08	12.84
DPLSM	-15.99	-34.06	-14.12	-16.52	-14.02	-33.64	-15.49

EXMPL #	2	30	58	86	114	142	170
DSH	19.31	5.55	14.86	12.58	15.71	11.75	19.37
DPL	1.30	-6.03	7.80	-2.94	7.17	-10.70	2.56
DSHSM	15.71	2.18	9.21	9.27	10.27	8.60	15.67
DPLSM	-6.13	-12.99	1.56	-9.45	1.17	-16.90	-5.08

EXMPL #	3	31	59	87	115	143	171
DSH	12.20	-0.99	10.82	12.09	12.43	-0.53	10.51
DPL	-1.12	-1.12	0.69	-1.25	-0.87	-0.53	0.34
DSHSM	3.14	-9.33	1.27	3.17	3.32	-8.69	1.00
DPLSM	-10.18	-9.45	-8.85	-10.17	-9.97	-8.69	-9.16

EXMPL #	4	32	60	88	116	144	172
DSH	0.30	0.19	0.68	0.09	0.17	-0.07	0.25
DPL	-6.82	-5.51	-5.62	-6.25	-4.90	-5.16	-8.65
DSHSM	-4.15	-3.38	-5.14	-4.37	-4.51	-3.65	-5.31
DPLSM	-13.50	-10.86	-13.17	-12.70	-10.97	-10.32	-17.00

EXMPL #	5	33	61	89	117	145	173
DSH	10.00	-4.99	8.60	10.32	9.82	-4.49	8.94
DPL	1.31	-4.99	2.07	-2.99	1.11	-4.49	2.43
DSHSM	4.09	-10.51	1.89	5.18	3.32	-9.28	2.85
DPLSM	-4.60	-10.51	-4.64	-8.13	-5.39	-9.28	-3.66

EXMPL #	6	34	62	90	118	146	174
DSH	1.29	0.26	4.31	2.55	0.77	0.27	6.14
DPL	-8.25	-7.45	2.04	-6.23	-1.11	-6.73	-5.19
DSHSM	-1.52	-2.01	-0.59	-0.22	-3.30	-2.00	2.81
DPLSM	-14.03	-12.13	-3.38	-11.67	-5.60	-11.18	-12.07

EXMPL #	7	35	63	91	119	147	175
DSH	0.13	-0.27	2.91	0.75	0.52	0.04	3.19
DPL	0.13	-0.27	2.91	0.75	0.52	0.04	3.19
DSHSM	-6.43	-5.53	-5.12	-5.65	-6.06	-5.12	-4.76
DPLSM	-6.43	-5.53	-5.12	-5.65	-6.06	-5.12	-4.76

EXMPL #	8	36	64	92	120	148	176
DSH	0.54	0.38	6.24	1.18	0.46	0.15	6.37
DPL	0.54	0.38	6.24	1.18	0.46	0.15	6.37
DSHSM	-4.25	-3.46	0.05	-2.97	-4.80	-3.21	0.73
DPLSM	-4.25	-3.46	0.05	-2.97	-4.80	-3.21	0.73

EXMPL #	9	37	65	93	121	149	177
DSH	17.64	2.18	12.61	10.19	13.26	9.77	17.52
DPL	4.77	-6.14	9.44	-0.78	8.92	2.80	5.58
DSHSM	15.07	-0.27	8.06	7.85	8.83	7.51	14.89
DPLSM	-0.53	-11.18	4.89	-5.37	4.49	-1.63	0.16

EXMPL #	10	38	66	94	122	150	178
DSH	10.37	-4.27	9.53	10.35	10.18	-3.29	9.38
DPL	0.85	-4.27	2.40	0.83	0.63	-3.29	2.24
DSHSM	2.15	-11.92	1.03	2.23	1.91	-10.78	0.91
DPLSM	-7.37	-11.92	-6.10	-7.30	-7.63	-10.78	-6.23

EXMPL #	11	39	67	95	123	151	179
DSH	-0.23	-0.26	0.16	0.33	0.63	0.48	-0.09
DPL	-5.24	-4.27	-4.28	-4.10	-2.90	-3.06	-6.34
DSHSM	-3.37	-2.77	-3.94	-2.78	-2.63	-2.01	-4.00
DPLSM	-9.94	-8.03	-9.59	-8.60	-7.13	-6.65	-12.21

EXMPL #	12	40	68	96	124	152	180
DSH	7.90	-8.55	7.21	7.79	7.77	-8.16	7.45
DPL	2.14	-8.55	2.94	-1.07	2.01	-8.16	3.20
DSHSM	2.93	-13.24	1.68	3.49	2.34	-12.19	2.41
DPLSM	-2.83	-13.24	-2.59	-5.37	-3.42	-12.19	-1.85

EXMPL #	13	41	69	97	125	153	181
DSH	0.31	-0.04	1.52	0.86	-0.21	0.06	2.48
DPL	-5.92	-5.04	1.52	-4.77	-0.21	-4.48	-5.14
DSHSM	-1.53	-1.51	-2.18	-0.97	-3.22	-1.41	0.23
DPLSM	-9.70	-8.08	-2.18	-8.35	-3.22	-7.37	-9.77

EXMPL #	14	42	70	98	126	154	182
DSH	0.29	0.18	0.75	0.33	0.03	0.00	1.19
DPL	0.29	0.18	0.75	0.33	0.03	0.00	1.19
DSHSM	-5.09	-4.12	-5.97	-4.99	-5.39	-4.26	-5.47
DPLSM	-5.09	-4.12	-5.97	-4.99	-5.39	-4.26	-5.47

EXMPL #	15	43	71	99	127	155	183
DSH	0.98	-0.40	8.33	0.95	0.95	-0.33	8.42
DPL	0.98	-0.40	7.23	0.76	0.95	-0.49	7.32
DSHSM	-2.99	-3.62	3.06	-2.11	-3.61	-2.81	3.82
DPLSM	-2.99	-3.62	1.97	-2.34	-3.61	-3.00	2.73

EXMPL #	16	44	72	100	128	156	184
DSH	1.00	0.01	8.02	0.95	0.74	-0.22	8.18
DPL	1.00	0.01	6.92	0.95	0.74	-0.22	7.09
DSHSM	-4.04	-4.06	1.81	-3.71	-4.63	-3.99	2.34
DPLSM	-4.04	-4.06	0.71	-3.71	-4.63	-3.99	1.24

EXMPL #	17	45	73	101	129	157	185
DSH	0.26	0.27	0.88	-0.07	0.29	0.07	1.60
DPL	0.26	0.27	0.88	-0.07	0.29	0.07	1.60
DSHSM	-3.74	-2.93	-4.57	-3.58	-4.10	-2.73	-3.34
DPLSM	-3.74	-2.93	-4.57	-3.58	-4.10	-2.73	-3.34

EXMPL #	18	46	74	102	130	158	186
DSH	1.07	-0.15	10.10	1.07	0.98	-0.44	9.46
DPL	1.07	-0.15	5.97	1.07	0.98	-0.44	5.30
DSHSM	-3.25	-3.65	4.74	-2.67	-3.74	-3.47	4.52
DPLSM	-3.25	-3.65	0.61	-2.67	-3.74	-3.47	0.36

EXMPL #	19	47	75	103	131	159	187
DSH	0.17	-0.20	1.95	0.18	0.08	-0.06	2.41
DPL	0.17	-0.20	1.95	0.02	0.08	-0.19	2.41
DSHSM	-3.26	-2.96	-2.88	-2.46	-3.86	-2.18	-1.79
DPLSM	-3.26	-2.96	-2.88	-2.66	-3.86	-2.34	-1.79

EXMPL #	20	48	76	104	132	160	188
DSH	0.20	-0.14	2.26	0.00	0.66	-0.11	2.37
DPL	0.20	-0.14	2.26	0.00	0.66	-0.11	2.37
DSHSM	-4.16	-3.64	-3.40	-4.03	-3.94	-3.34	-2.96
DPLSM	-4.16	-3.64	-3.40	-4.03	-3.94	-3.34	-2.96

EXMPL #	21	49	77	105	133	161	189
DSH	-0.03	0.05	3.23	-0.11	0.02	-0.19	3.50
DPL	-0.03	0.05	3.23	-0.11	0.02	-0.19	3.50
DSHSM	-3.85	-3.00	-1.82	-3.42	-4.16	-2.83	-1.10
DPLSM	-3.85	-3.00	-1.82	-3.42	-4.16	-2.83	-1.10

EXMPL #	22	50	78	106	134	162	190
DSH	0.61	0.03	4.31	0.23	0.46	-0.18	4.20
DPL	0.61	0.03	4.31	0.23	0.46	-0.18	4.20
DSHSM	-3.45	-3.23	-1.06	-3.26	-4.01	-2.99	-0.68
DPLSM	-3.45	-3.23	-1.06	-3.26	-4.01	-2.99	-0.68

EXMPL #	23	51	79	107	135	163	191
DSH	0.08	-0.01	2.77	-0.01	0.45	0.02	2.92
DPL	0.08	-0.01	2.77	-0.01	0.45	0.02	2.92
DSHSM	-3.90	-3.20	-2.50	-3.51	-3.87	-2.77	-1.91
DPLSM	-3.90	-3.20	-2.50	-3.51	-3.87	-2.77	-1.91

EXMPL #	24	52	80	108	136	164	192
DSH	0.29	0.07	2.86	0.05	0.16	0.30	3.14
DPL	0.29	0.07	2.86	0.05	0.16	0.30	3.14
DSHSM	-3.42	-2.91	-2.14	-3.09	-3.95	-2.21	-1.37
DPLSM	-3.42	-2.91	-2.14	-3.09	-3.95	-2.21	-1.37

EXMPL #	25	53	81	109	137	165	193
DSH	0.11	0.22	4.99	0.27	0.38	-0.11	5.23
DPL	0.11	0.22	4.99	0.27	0.38	-0.11	5.08
DSHSM	-3.87	-2.97	-0.18	-3.16	-3.96	-2.86	0.50
DPLSM	-3.87	-2.97	-0.18	-3.16	-3.96	-2.86	0.36

EXMPL #	26	54	82	110	138	166	194
DSH	0.18	-0.09	2.52	0.30	0.26	0.02	2.74
DPL	0.18	-0.09	2.52	0.30	0.26	0.02	2.74
DSHSM	-3.71	-3.21	-2.70	-3.02	-4.02	-2.65	-1.99
DPLSM	-3.71	-3.21	-2.70	-3.02	-4.02	-2.65	-1.99

EXMPL #	27	55	83	111	139	167	195
DSH	0.32	-0.13	4.48	0.42	0.35	0.08	4.80
DPL	0.32	-0.13	4.48	0.42	0.35	0.08	4.80
DSHSM	-3.83	-3.47	-0.94	-3.21	-4.17	-2.84	-0.16
DPLSM	-3.83	-3.47	-0.94	-3.21	-4.17	-2.84	-0.16

EXMPL #	28	56	84	112	140	168	196
DSH	-0.04	-0.18	4.59	0.24	0.35	-0.13	4.66
DPL	-0.04	-0.18	4.59	0.24	0.35	-0.13	4.66
DSHSM	-3.94	-3.30	-0.54	-3.03	-3.94	-2.76	0.02
DPLSM	-3.94	-3.30	-0.54	-3.03	-3.94	-2.76	0.02

D.2 Percentage Difference Between the MEANS of the MONTE CARLO Simulation and the DSH, DPL, DSHSM and DPLSM Heuristics for the 8 Queue Model 45 Examples

EXMPL #	197	213	229	245	261	277	293
DSH	-0.05	-0.02	-0.01	0.35	-0.05	0.02	0.05
DPL	-0.05	-0.02	-0.01	0.35	-0.05	0.02	0.05
DSHSM	-2.00	-2.07	-1.85	-1.90	-1.73	-1.75	-2.07
DPLSM	-2.00	-2.07	-1.85	-1.90	-1.73	-1.75	-2.07

EXMPL #	198	214	230	246	262	278	294
DSH	0.08	0.27	0.03	1.70	-0.04	-0.03	0.42
DPL	0.08	-0.50	0.03	1.68	-0.04	-0.03	-0.34
DSHSM	-2.02	-1.92	-1.96	-0.71	-1.85	-1.93	-1.85
DPLSM	-2.02	-2.69	-1.96	-0.73	-1.85	-1.93	-2.61

EXMPL #	199	215	231	247	263	279	295
DSH	0.01	0.07	0.00	0.25	0.06	-0.03	0.01
DPL	0.01	0.07	0.00	0.25	0.06	-0.03	0.01
DSHSM	-2.10	-2.15	-1.97	-2.17	-1.75	-1.94	-2.27
DPLSM	-2.10	-2.15	-1.97	-2.17	-1.75	-1.94	-2.27

EXMPL #	200	216	232	248	264	280	296
DSH	-0.05	0.03	0.11	0.21	-0.03	0.09	-0.05
DPL	-0.05	0.03	0.11	0.21	-0.03	0.09	-0.05
DSHSM	-2.03	-2.03	-1.76	-2.08	-1.72	-1.69	-2.18
DPLSM	-2.03	-2.03	-1.76	-2.08	-1.72	-1.69	-2.18

EXMPL #	201	217	233	249	265	281	297
DSH	-0.01	0.11	0.04	0.91	-0.01	-0.01	0.19
DPL	-0.01	0.11	0.04	0.91	-0.01	-0.01	0.19
DSHSM	-2.15	-2.15	-1.97	-1.55	-1.85	-1.96	-2.14
DPLSM	-2.15	-2.15	-1.97	-1.55	-1.85	-1.96	-2.14

EXMPL #	202	218	234	250	266	282	298
DSH	-0.08	0.12	0.02	0.24	0.10	-0.03	0.02
DPL	-0.08	0.12	0.02	0.24	-0.44	-0.03	0.02
DSHSM	-2.28	-2.20	-2.03	-2.27	-1.80	-2.00	-2.35
DPLSM	-2.28	-2.20	-2.03	-2.27	-2.36	-2.00	-2.35

EXMPL #	203	219	235	251	267	283	299
DSH	-0.09	0.05	-0.11	0.62	-0.03	-0.08	-0.06
DPL	-0.09	0.05	-0.11	0.62	-0.03	-0.08	-0.06
DSHSM	-2.37	-2.37	-2.22	-1.98	-1.99	-2.15	-2.55
DPLSM	-2.37	-2.37	-2.22	-1.98	-1.99	-2.15	-2.55

EXMPL #	204	220	236	252	268	284	300
DSH	0.05	0.01	-0.06	0.26	0.06	-0.02	0.06
DPL	0.05	0.01	-0.06	0.26	0.06	-0.02	0.06
DSHSM	-1.93	-2.05	-1.92	-2.03	-1.62	-1.80	-2.09
DPLSM	-1.93	-2.05	-1.92	-2.03	-1.62	-1.80	-2.09

EXMPL #	205	221	237	253	269	285	301
DSH	0.03	-0.02	0.07	0.24	-0.07	-0.07	0.01
DPL	0.03	-0.02	0.07	0.24	-0.07	-0.07	0.01
DSHSM	-2.04	-2.19	-1.87	-2.14	-1.85	-1.94	-2.22
DPLSM	-2.04	-2.19	-1.87	-2.14	-1.85	-1.94	-2.22

EXMPL #	206	222	238	254	270	286	302
DSH	0.16	0.07	0.09	0.80	-0.02	-0.01	0.14
DPL	0.16	0.07	0.09	0.80	-0.14	-0.01	0.14
DSHSM	-1.91	-2.09	-1.86	-1.57	-1.80	-1.86	-2.08
DPLSM	-1.91	-2.09	-1.86	-1.57	-1.93	-1.86	-2.08

EXMPL #	207	223	239	255	271	287	303
DSH	0.02	0.14	0.02	1.57	0.02	-0.09	0.37
DPL	0.02	-0.52	0.02	1.57	0.02	-0.09	-1.42
DSHSM	-2.14	-2.12	-2.00	-0.88	-1.83	-2.04	-1.97
DPLSM	-2.14	-2.78	-2.00	-0.88	-1.83	-2.04	-3.76

EXMPL #	208	224	240	256	272	288	304
DSH	0.02	-0.05	0.10	0.60	0.03	-0.01	-0.01
DPL	0.02	-0.05	0.10	0.60	0.03	-0.01	-0.01
DSHSM	-1.99	-2.15	-1.78	-1.72	-1.69	-1.82	-2.19
DPLSM	-1.99	-2.15	-1.78	-1.72	-1.69	-1.82	-2.19

EXMPL #	209	225	241	257	273	289	305
DSH	-0.01	0.06	-0.01	0.23	-0.05	0.03	-0.06
DPL	-0.01	0.06	-0.01	0.23	-0.05	0.03	-0.06
DSHSM	-2.05	-2.06	-1.93	-2.13	-1.79	-1.80	-2.26
DPLSM	-2.05	-2.06	-1.93	-2.13	-1.79	-1.80	-2.26

EXMPL #	210	226	242	258	274	290	306
DSH	0.24	0.52	0.09	4.41	0.02	0.10	0.58
DPL	0.24	0.52	0.09	1.13	0.02	0.10	-1.27
DSHSM	-2.52	-2.36	-2.50	1.19	-2.35	-2.39	-2.40
DPLSM	-2.52	-2.36	-2.50	-2.08	-2.35	-2.39	-4.25

EXMPL #	211	227	243	259	275	291	307
DSH	-0.05	0.32	0.03	1.32	-0.09	0.21	0.28
DPL	-0.05	0.32	0.03	1.32	-0.09	0.21	-0.42
DSHSM	-2.62	-2.40	-2.37	-1.61	-2.31	-2.13	-2.51
DPLSM	-2.62	-2.40	-2.37	-1.61	-2.31	-2.13	-3.21

EXMPL #	212	228	244	260	276	292	308
DSH	0.04	0.07	0.08	0.56	0.06	0.06	0.05
DPL	0.04	0.07	0.08	0.56	0.06	0.06	0.05
DSHSM	-2.00	-2.07	-1.84	-1.79	-1.70	-1.79	-2.17
DPLSM	-2.00	-2.07	-1.84	-1.79	-1.70	-1.79	-2.17

D.3 Percentage Difference Between the STANDARD DEVIATION of the MONTE CARLO Simulation and the DSH, DPL, DSHSM and DPLSM Heuristics for the 4 Queue Model 25 Examples

EXMPL #	1	29	57	85	113	141	169
DSH	-2.72	-3.80	-4.18	-7.26	-0.24	-5.36	-2.50
DPL	-18.61	-22.94	-16.86	-20.64	-14.34	-26.35	-16.74
DSHSM	-3.05	-4.04	-1.03	-7.55	2.40	-5.59	-2.88
DPLSM	-10.95	-17.60	-10.00	-14.54	-8.70	-21.87	-7.88

EXMPL #	2	30	58	86	114	142	170
DSH	-8.31	-9.62	-6.05	-12.67	-4.72	-10.59	-7.64
DPL	-19.78	-15.28	-10.01	-20.79	-10.15	-22.68	-18.05
DSHSM	-8.46	-9.73	-4.59	-12.79	-3.36	-10.69	-7.79
DPLSM	-12.11	-9.25	-10.75	-15.51	-10.83	-18.09	-9.92

EXMPL #	3	31	59	87	115	143	171
DSH	-8.99	-1.10	-10.26	-10.63	-5.38	-4.32	-13.77
DPL	-16.97	-0.77	-15.39	-18.40	-13.27	-2.63	-18.95
DSHSM	-2.22	3.52	-1.31	-5.17	2.71	-0.48	-6.31
DPLSM	-10.69	3.87	-6.87	-13.31	-5.79	1.27	-11.83

EXMPL #	4	32	60	88	116	144	172
DSH	0.79	1.30	-5.01	-4.67	0.92	1.24	-7.00
DPL	-2.69	-1.04	-8.89	-7.81	-1.62	-0.79	-12.35
DSHSM	0.65	1.21	-3.35	-4.81	2.00	1.15	-7.22
DPLSM	0.79	1.30	-5.01	-4.67	0.92	1.24	-7.00

EXMPL #	5	33	61	89	117	145	173
DSH	-12.42	-1.66	-8.95	-22.99	-5.54	-7.39	-12.69
DPL	-17.72	-1.66	-12.32	-22.99	-10.90	-7.39	-15.91
DSHSM	-8.02	1.30	-2.74	-20.70	0.26	-5.84	-8.13
DPLSM	-13.53	1.30	-6.30	-20.70	-5.39	-5.84	-11.49

EXMPL #	6	34	62	90	118	146	174
DSH	-5.30	3.39	-16.03	-15.95	2.58	-0.29	-24.65
DPL	-10.27	0.17	-17.27	-20.32	1.82	-3.11	-32.45
DSHSM	-5.39	3.33	-14.78	-16.04	3.35	-0.35	-24.80
DPLSM	-5.30	3.39	-17.92	-15.95	1.42	-0.29	-24.65

EXMPL #	7	35	63	91	119	147	175
DSH	-1.79	4.72	-17.09	-4.06	-1.80	3.92	-17.95
DPL	-1.79	4.72	-17.09	-4.06	-1.80	3.92	-17.95
DSHSM	2.19	7.32	-9.54	-0.77	3.06	6.06	-11.76
DPLSM	2.19	7.32	-9.54	-0.77	3.06	6.06	-11.76

EXMPL #	8	36	64	92	120	148	176
DSH	-5.78	6.42	-19.47	-8.92	-1.88	1.66	-23.23
DPL	-5.78	6.42	-17.56	-8.92	-1.88	1.66	-23.23
DSHSM	-2.77	8.33	-13.28	-7.39	2.12	2.68	-18.66
DPLSM	-2.77	8.33	-11.26	-7.39	2.12	2.68	-18.66

EXMPL #	9	37	65	93	121	149	177
DSH	-9.03	-10.39	-10.10	-12.58	-6.89	-12.50	-11.64
DPL	-17.24	-9.61	-8.01	-18.14	-6.28	-14.86	-19.19
DSHSM	-9.14	-10.47	-7.78	-12.66	-4.73	-12.57	-11.75
DPLSM	-11.72	-5.10	-9.68	-14.51	-7.81	-11.35	-13.28

EXMPL #	10	38	66	94	122	150	178
DSH	-10.58	-2.25	-11.38	-12.61	-10.45	-2.30	-12.33
DPL	-16.30	2.65	-14.95	-18.27	-16.26	2.48	-15.87
DSHSM	-1.32	4.08	-0.46	-4.30	-0.04	3.33	-2.56
DPLSM	-7.54	9.31	-4.41	-10.39	-6.42	8.41	-6.43

EXMPL #	11	39	67	95	123	151	179
DSH	-0.82	1.84	-3.56	-1.10	1.48	0.79	-3.59
DPL	-3.31	0.23	-6.30	-3.26	-0.29	-0.63	-7.33
DSHSM	-0.92	1.77	-2.40	-1.20	2.22	0.73	-3.74
DPLSM	-0.82	1.84	-3.56	-1.10	1.48	0.79	-3.59

EXMPL #	12	40	68	96	124	152	180
DSH	-10.50	3.23	-9.05	-24.00	-9.25	-3.11	-12.96
DPL	-13.91	3.23	-11.22	-24.00	-12.89	-3.11	-15.04
DSHSM	-5.16	6.74	-1.39	-20.48	-1.77	-0.84	-7.38
DPLSM	-8.73	6.74	-3.72	-20.48	-5.66	-0.84	-9.56

EXMPL #	13	41	69	97	125	153	181
DSH	-2.70	3.96	-6.98	-12.72	1.74	-0.90	-20.51
DPL	-5.86	1.89	-4.13	-15.47	3.63	-2.74	-25.46
DSHSM	-2.76	3.92	-5.33	-12.77	2.84	-0.94	-20.61
DPLSM	-2.70	3.96	-5.33	-12.72	2.84	-0.90	-20.51

EXMPL #	14	42	70	98	126	154	182
DSH	-0.95	3.75	-15.93	0.08	-1.54	4.69	-13.00
DPL	-0.95	3.75	-15.93	0.08	-1.54	4.69	-13.00
DSHSM	3.93	6.98	-7.44	4.40	3.94	7.57	-5.61
DPLSM	3.93	6.98	-7.44	4.40	3.94	7.57	-5.61

EXMPL #	15	43	71	99	127	155	183
DSH	-5.64	6.36	-17.84	-14.03	-2.37	1.36	-25.77
DPL	-5.64	6.36	-18.47	-14.03	-2.37	1.36	-26.38
DSHSM	-3.14	7.96	-11.71	-12.99	1.70	2.03	-21.95
DPLSM	-3.14	7.96	-12.37	-13.16	1.70	1.92	-22.58

EXMPL #	16	44	72	100	128	156	184
DSH	-5.26	6.66	-21.72	-9.68	-3.83	4.84	-23.53
DPL	-5.26	6.66	-22.35	-9.68	-3.83	4.84	-24.13
DSHSM	-0.46	9.71	-12.88	-6.24	1.96	7.04	-16.30
DPLSM	-0.46	9.71	-13.55	-6.24	1.96	7.04	-16.94

EXMPL #	17	45	73	101	129	157	185
DSH	-0.67	1.60	-11.24	-4.69	1.86	2.95	-15.95
DPL	-0.67	1.60	-11.24	-4.69	1.86	2.95	-15.95
DSHSM	1.71	3.23	-6.37	-3.44	5.03	3.78	-12.28
DPLSM	1.71	3.23	-6.37	-3.44	5.03	3.78	-12.28

EXMPL #	18	46	74	102	130	158	186
DSH	-6.48	9.11	-18.02	-13.17	-3.32	5.09	-22.28
DPL	-6.48	9.11	-20.37	-13.17	-3.32	5.09	-24.53
DSHSM	-2.34	11.67	-9.73	-10.51	2.28	6.76	-16.24
DPLSM	-2.34	11.67	-12.26	-10.51	2.28	6.76	-18.61

EXMPL #	19	47	75	103	131	159	187
DSH	0.02	3.59	-16.62	-6.62	3.07	1.42	-21.74
DPL	0.02	3.59	-16.62	-6.62	3.07	1.42	-21.74
DSHSM	2.05	4.98	-11.36	-5.77	6.32	2.00	-18.50
DPLSM	2.05	4.98	-11.36	-5.91	6.32	1.90	-18.50

EXMPL #	20	48	76	104	132	160	188
DSH	-1.09	5.24	-17.94	-4.22	-0.25	3.39	-20.44
DPL	-1.09	5.24	-17.94	-4.22	-0.25	3.39	-20.44
DSHSM	2.84	7.85	-10.49	-1.40	4.50	5.28	-14.26
DPLSM	2.84	7.85	-10.49	-1.40	4.50	5.28	-14.26

EXMPL #	21	49	77	105	133	161	189
DSH	-2.75	7.30	-20.67	-6.67	1.46	0.07	-21.92
DPL	-2.75	7.30	-20.67	-6.67	1.46	0.07	-21.92
DSHSM	0.74	9.55	-13.17	-4.45	6.08	1.60	-16.54
DPLSM	0.74	9.55	-13.17	-4.45	6.08	1.60	-16.54

EXMPL #	22	50	78	106	134	162	190
DSH	-1.43	6.34	-19.52	-6.56	0.21	2.39	-23.00
DPL	-1.43	6.34	-15.57	-6.56	0.21	2.39	-23.00
DSHSM	1.87	8.52	-12.81	-4.29	4.45	3.92	-17.80
DPLSM	1.87	8.52	-8.62	-4.29	4.45	3.92	-17.80

EXMPL #	23	51	79	107	135	163	191
DSH	-0.65	6.72	-20.69	-6.91	1.59	2.96	-21.47
DPL	-0.65	6.72	-20.69	-6.91	1.59	2.96	-21.47
DSHSM	2.92	9.07	-13.51	-4.27	6.00	4.70	-15.84
DPLSM	2.92	9.07	-13.51	-4.27	6.00	4.70	-15.84

EXMPL #	24	52	80	108	136	164	192
DSH	-0.96	3.53	-19.89	-8.05	1.22	0.55	-24.63
DPL	-0.96	3.53	-19.89	-8.05	1.22	0.55	-24.63
DSHSM	2.04	5.57	-13.07	-6.25	5.44	1.76	-19.81
DPLSM	2.04	5.57	-13.07	-6.25	5.44	1.76	-19.81

EXMPL #	25	53	81	109	137	165	193
DSH	-2.03	7.10	-22.73	-6.38	-0.42	1.09	-24.20
DPL	-2.03	7.10	-20.27	-6.38	-0.42	1.09	-24.20
DSHSM	1.59	9.46	-14.81	-4.08	4.51	2.67	-18.51
DPLSM	1.59	9.46	-12.16	-4.08	4.51	2.67	-18.51

EXMPL #	26	54	82	110	138	166	194
DSH	-0.46	5.07	-17.32	-4.72	3.53	3.94	-22.66
DPL	-0.46	5.07	-17.32	-4.72	3.53	3.94	-22.66
DSHSM	2.66	7.16	-11.00	-2.59	7.42	5.38	-17.67
DPLSM	2.66	7.16	-11.00	-2.59	7.42	5.38	-17.67

EXMPL #	27	55	83	111	139	167	195
DSH	-3.22	4.89	-21.42	-6.68	0.68	2.97	-24.63
DPL	-3.22	4.89	-18.25	-6.68	0.68	2.97	-24.63
DSHSM	0.62	7.41	-13.91	-3.94	5.35	4.80	-18.63
DPLSM	0.62	7.41	-10.52	-3.94	5.35	4.80	-18.63

EXMPL #	28	56	84	112	140	168	196
DSH	-3.80	6.92	-19.34	-9.53	0.43	1.97	-27.44
DPL	-3.80	6.92	-16.19	-9.53	0.43	1.97	-27.44
DSHSM	-0.58	8.98	-12.27	-7.63	4.90	3.22	-22.32
DPLSM	-0.58	8.98	-8.92	-7.63	4.90	3.22	-22.32

D.4 Percentage Difference Between the STANDARD DEVIATION of the MONTE CARLO Simulation and the DSH, DPL, DSHSM and DPLSM Heuristics for the 8 Queue Model 45 Examples

EXMPL #	197	213	229	245	261	277	293
DSH	1.74	-5.59	4.27	-13.52	4.47	-0.22	-6.65
DPL	1.74	-5.59	4.27	-13.52	4.47	-0.22	-6.65
DSHSM	10.55	3.57	12.72	-1.01	11.21	7.35	3.28
DPLSM	10.55	3.57	12.72	-1.01	11.21	7.35	3.28

EXMPL #	198	214	230	246	262	278	294
DSH	-1.18	-14.21	6.39	-26.65	4.07	0.97	-14.07
DPL	-1.18	-14.21	6.39	-26.65	4.07	0.97	-14.07
DSHSM	8.05	-4.30	14.89	-12.52	10.97	8.52	-3.44
DPLSM	8.05	-4.30	14.89	-12.52	10.97	8.52	-3.44

EXMPL #	199	215	231	247	263	279	295
DSH	0.70	-4.56	5.81	-10.44	3.38	2.36	-5.36
DPL	0.70	-4.56	5.81	-10.44	3.38	2.36	-5.36
DSHSM	9.76	4.87	14.16	1.82	10.44	9.93	4.82
DPLSM	9.76	4.87	14.16	1.82	10.44	9.93	4.82

EXMPL #	200	216	232	248	264	280	296
DSH	0.57	-5.75	5.13	-7.29	3.08	-0.56	-8.20
DPL	0.57	-5.75	5.13	-7.29	3.08	-0.56	-8.20
DSHSM	8.87	2.76	13.00	3.75	9.37	6.43	1.19
DPLSM	8.87	2.76	13.00	3.75	9.37	6.43	1.19

EXMPL #	201	217	233	249	265	281	297
DSH	-0.96	-10.25	3.37	-21.51	4.04	1.48	-10.00
DPL	-0.96	-10.25	3.37	-21.51	4.04	1.48	-10.00
DSHSM	8.63	0.03	12.36	-7.53	11.29	9.39	0.96
DPLSM	8.63	0.03	12.36	-7.53	11.29	9.39	0.96

EXMPL #	202	218	234	250	266	282	298
DSH	0.47	-8.08	6.36	-9.27	2.46	3.40	-4.79
DPL	0.47	-8.08	6.36	-9.27	2.46	3.40	-4.79
DSHSM	9.08	1.09	14.30	2.20	9.13	10.40	4.72
DPLSM	9.08	1.09	14.30	2.20	8.75	10.40	4.72

EXMPL #	203	219	235	251	267	283	299
DSH	1.26	-6.42	4.74	-18.35	4.92	0.92	-9.35
DPL	1.26	-6.42	4.74	-15.18	4.92	0.92	-8.82
DSHSM	10.81	3.89	13.59	-4.51	12.34	9.12	1.74
DPLSM	10.81	3.89	13.59	-4.51	12.34	9.12	1.74

EXMPL #	204	220	236	252	268	284	300
DSH	1.37	-6.47	6.49	-9.76	6.35	1.09	-10.15
DPL	1.37	-6.47	6.49	-6.57	6.35	1.09	-9.41
DSHSM	9.76	2.20	14.37	1.81	12.54	8.15	-0.40
DPLSM	9.76	2.20	14.37	1.81	12.54	8.15	-0.40

EXMPL #	205	221	237	253	269	285	301
DSH	0.60	-5.79	5.25	-9.09	5.41	2.68	-6.27
DPL	0.60	-5.79	5.25	-9.09	5.41	2.68	-6.27
DSHSM	9.30	3.25	13.39	2.60	11.94	9.79	3.44
DPLSM	9.30	3.25	13.39	2.60	11.94	9.79	3.44

EXMPL #	206	222	238	254	270	286	302
DSH	-0.11	-9.06	5.58	-20.45	3.70	0.95	-9.92
DPL	-0.11	-9.06	5.58	-20.45	3.70	0.95	-9.92
DSHSM	8.47	-0.11	13.70	-7.67	10.13	7.97	-0.23
DPLSM	8.47	-0.11	13.70	-7.67	10.04	7.97	-0.23

EXMPL #	207	223	239	255	271	287	303
DSH	1.30	-14.66	6.25	-29.37	4.34	1.12	-17.38
DPL	1.30	-14.66	6.25	-25.08	4.34	1.12	-16.28
DSHSM	10.52	-4.26	14.92	-14.65	11.43	8.89	-6.09
DPLSM	10.52	-4.26	14.92	-14.65	11.43	8.89	-6.09

EXMPL #	208	224	240	256	272	288	304
DSH	1.86	-7.53	6.47	-16.30	4.35	2.46	-7.87
DPL	1.86	-7.53	6.47	-11.51	4.35	2.46	-6.03
DSHSM	10.56	1.67	14.72	-3.63	10.98	9.71	2.02
DPLSM	10.56	1.67	14.72	-3.63	10.98	9.71	2.02

EXMPL #	209	225	241	257	273	289	305
DSH	1.77	-7.10	5.29	-8.35	5.50	2.78	-3.60
DPL	1.77	-7.10	5.29	-8.35	5.50	2.78	-3.60
DSHSM	9.82	1.38	12.97	2.68	11.54	9.39	5.16
DPLSM	9.82	1.38	12.97	2.68	11.54	9.39	5.16

EXMPL #	210	226	242	258	274	290	306
DSH	-0.76	-13.60	6.26	-21.61	4.48	0.57	-19.39
DPL	-0.76	-13.60	6.26	-23.39	4.48	0.57	-18.76
DSHSM	7.82	-4.17	14.14	-7.34	10.94	7.69	-8.87
DPLSM	7.82	-4.17	14.14	-9.36	10.94	7.69	-8.87

EXMPL #	211	227	243	259	275	291	307
DSH	-1.46	-10.91	6.46	-25.03	4.60	2.43	-12.71
DPL	-1.46	-10.91	6.46	-25.03	4.60	2.43	-12.71
DSHSM	8.09	-0.62	14.99	-10.78	11.84	10.17	-1.49
DPLSM	8.09	-0.62	14.99	-10.78	11.84	10.17	-1.49

EXMPL #	212	228	244	260	276	292	308
DSH	0.87	-9.25	6.09	-17.87	4.33	1.58	-10.76
DPL	0.87	-9.25	6.09	-17.87	4.33	1.58	-10.76
DSHSM	10.01	0.54	14.64	-4.62	11.33	9.23	-0.15
DPLSM	10.01	0.54	14.64	-4.62	11.33	9.23	-0.15

BIBLIOGRAPHY

[1] Baskett, F., Chandy, K.M., Muntz, R. and Palacious, F.G., "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," *JACM*, 22, pp. 248-260 (1975).

[2] Boxma, O.J. and Donk P., "On Response Time and Cycle Time Distributions in a Two-Stage Cyclic Queue," *Performance Eval.*, 2, pp. 181-194 (1982).

[3] Boxma, O.J., Kelly, F.P. and Konheim A.G., "The Product Form for the Sojourn Time Distribution in Cyclic Exponential Queues," *JACM*, 31, pp. 128-133 (1984).

[4] Bulkin, M.H., Colley, J.L. and Steinhoff, H.W., "Load Forecasting, Priority Sequencing, and Simulation in a Job Shop Control System," *Management Science*, 13, pp. B29-B51 (1966).

[5] Burke, P.J., "The Output of a Queueing System," *Operations Research*, 4, pp. 699-704 (1956).

[6] Burke P.J., "Output Processes and Tandem Queues," *Proc. Symp. Comp.-Comm. Networks and Teletraffic*, J. Fox, ed., John Wiley and Sons, New York, pp. 419-428 (1972).

[7] Buzen, J.P., "Computational Algorithms for Closed Queueing Networks with Exponential Servers," *CACM*, 16, pp. 527-531 (1973).

[8] Chandy, K.M., "The Analysis of Solution of General Queueing Networks," *Proceedings of Sixth Annual Princeton Conference on Information Science and Systems*, Princeton University, Princeton, NJ, pp. 224- 228 (1972).

[9] Chandy, K.M., Herzog,U. and Woo,L., "Approximate Analysis of General Queueing Networks," *IBM J. Res. Devel.* 19, pp. 43-49 (1975).

[10] Chow, W., "The Cycle Time Distribution of Exponential Cyclic Queues," *JACM*, 27, pp. 281-286 (1980).

[11] Cohen J.W., *The Single Server Queue*, American Elsevier, New York (1969).

[12] Courtois, P.J., "Decomposability, Instabilities, and Saturation in Multiprogramming Systems," *CACM*, 18, pp. 371-377 (1975).

[13] Daduna, H., "Burke's Theorem on Passage Times in Gordon- Newell Networks," *Adv. Appl.Prob.*, 16, pp. 867-886 (1984).

[14] Daduna, H., "Passage Times for Overtake-free Paths in Gordon-Newell Networks," *Adv. Appl. Prob.*, 14, pp. 672-686 (1982).

[15] Disney R. and Cherry P., *Some Topics in Queueing Network Theory, Lecture Notes in Econ. and Math Systems No. 98*, Springer-Verlag, New York, pp. 23-44 (1974).

[16] Fayolle, G., Iasnogorodski, R. and Mitrani, I., "The Distribution of Sojourn Times in a Queueing Network with Overtaking: Reduction to a Boundary Value Problem," *Performance '83*, AK Agrawaluad S.K. Tripathi (Eds) North Holland Pub. Comp., pp. 477-486 (1983).

[17] Feller, W., *An Introduction to Probability Theory and Its Applications Vol.1*, John Wiley and Sons, Inc., New York (1950).

[18] Gaver, D.P., "Diffusion Approximations and Models for Certain Congestion Problems," *J. Appl. Prob.*, 5, pp. 607- 623 (1968).

[19] Gaver, D.P., "Analysis of Remote Technical Backlogs under heavy demand conditions," *JACM*, 18, pp. 405-415 (1971).

[20] Gaver, D.P. and Shedler, G.S., "Approximate Models for Processor Utilization in Multiprogrammed Computer Systems," IBM Res. Rep., RJ-1102 (1972).

[21] Giffin, W.C., *Queueing: Basic Theory and Application*, GRID Inc., Columbus, Ohio (1978).

[22] Gordon, W. and Newell, G., "Closed Queueing Systems with Exponential Servers," *Operations Research*, 15 , pp. 254-265 (1967).

[23] Grassman, W., "Transient Solutions in Markovian Queues," *Eur. J. of Operations Research*, 1, pp. 392-402 (1977).

[24] Grassman, W.K., "Transient Solutions in Markovian Queueing Systems," *Comp. Operations Research*, 4, pp. 47-56 (1977).

[25] Harrison, J.M., "Queueing Models for Assembly-Like Queues," Tech Rep No. 20, Dept of Operations Research, Stanford Univ., Stanford, California (1970).

[26] Harrison, J.M., "Assembly-like Queues," *J. Appl. Prob.*, 10, pp. 354-367 (1973).

[27] Harrison, J. M., "A Limit Theorem for Priority Queues in Heavy Traffic," *J. Appl. Prob.*, 10, pp. 907-912 (1973).

[28] Harrison, J.M., "The Heavy Traffic Approximation for Single Server Queues in Series," *J. Appl. Prob.*, 10, pp. 613-629 (1973).

[29] Harrison, J.M., "The Diffusion Approximation for Tandem Queues in Heavy Traffic," Tech. Rep No.75, Dept of Operations Research, Stanford Univ., Stanford, California (1977).

[30] Harrison, P.G., "Distributions of Time Delays in Queueing Networks," Unpublished Manuscript (1980).

[31] Harrison, P.G., "A Note on Cycle Times in Tree-like Queueing Networks," *Adv. Appl. Prob.*, 16, pp. 216-219 (1984).

[32] Iglehart, D.L. and Whitt, W., "Multiple Channel Queues in Heavy Traffic, I," *Adv. Appl. Prob.*, 2, pp. 150-177 (1970).

[33] Jackson, J.R., "Networks of Waiting Lines," *Operations Research*, 5, pp. 518-521 (1957).

[34] Jackson, J.R., "Job Shop-like Queueing Systems," *Management Science*, 10 pp. 131-142 (1963).

[35] Jackson, R.R.P., "Queueing Systems with Phase Type Service," *Operations Research Quarterly*, 5, pp. 109-120 (1954).

[36] Keilson, J., *Markov Chain Models: Rarity and Exponentiality*, Springer-Verlog, New York (1979).

[37] Kelly, F.P., "Networks of Queues," *Adv. Appl. Prob.*, 8, pp. 416-432 (1976).

[38] Kelly, F.P., "Networks of Queues with Customers of Different Types," *J. Appl. Prob.*, 12, pp. 542-554 (1975).

[39] Kelly, F.P. and Pollett,P.K., "Sojourn Times in Closed Queueing Networks," *Adv. Appl. Prob.*, 15, pp. 638-656 (1983).

[40] Kingman, J.F.C., "On Queues in Heavy Traffic," *J. Royal Stat. Soc., Series B*, 24, pp. 383-392 (1962).

[41] Kobayashi, H., "Applications of Diffusion Approximations to Queueing Networks I: Equilibrium Distributions," *JACM*, 21, pp. 316-328 (1974).

[42] Kobayashi, H., "Applications of the Diffusion Approximations to Queueing Networks. II. Nonequilibrium Distributions and Applications to Computer Modeling," *JACM*, 21, pp. 459-469 (1974).

[43] Kuehn, P.J., "Approximate Analysis of General Queueing Networks by Decomposition," *IEEE Trans. Com.*, COM-27, pp. 113-126 (1979).

[44] Lam, S.S. and Shankar, A.V., "Response Time Distributions for a Multi-Class Queue with Feedback," *Performance '80*, Toronto, pp. 225-234 (1980).

[45] Law, A.M. and Kelton, W.D., *Simulation Modeling and Analysis*, McGraw-Hill, Inc., New York (1982).

[46] Lemoine, A.J., "Networks of Queues – A Survey of Equilibrium Analysis," *Management Science*, 24, pp. 464-481 (1977).

[47] Lippman, S.A., "Applying a New Device in the Optimization of Exponential Queueing Systems," *Operations Research*, 23, pp. 687-710 (1975).

[48] Melamed, B., "Sojourn Times in Queueing Networks," *Math. of Operations Research*, 7, pp. 223-244 (1982).

[49] Melamed, B. and Yadin, M., "Numerical Computation of Sojourn-Time Distributions in Queueing Networks," *JACM*, 31, pp. 839-854 (1984).

[50] Melamed, B. and Yadin, M., "Randomization Procedures in the Computation of Cumulative-Time Distributions over Discrete State Markov Processes," *Operations Research*, 32, pp. 926-944 (1984).

[51] Mitrani, I., "A Critical Note on a Result by Lemoine," *Management Science*, 25, pp. 1026-1027 (1979).

[52] Muth, E.J., *Transform Methods with Applications to Engineering and Operations Research*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1977).

[53] Newell, G.F., *Applications of Queueing Theory*, Chapman and Hall, London (1971).

[54] Platzman, L.K., Ammons, J.C. and Bartholdi, J.J., "A Simple and Efficient Algorithm to Compute Tail Probabilities from Transforms," (*Operations Research* to appear).

[55] Rao, N.P., "On the Mean Production Rate of a Two State Production System of the Tandem Type," *Int. J. Prod. Res.*, 13, pp. 207-217 (1975).

[56] Reich, E., "Waiting Times when Queues are in Tandem," *Ann. Math. Stat.*, 28, pp. 768-773 (1957).

[57] Reich, E., "Note on Queueing in Tandem," *Ann. Math. Stat.*, 34, pp. 338-341 (1963).

[58] Reiman, M.I., "Queueing Networks in Heavy Traffic," Tech Rep No.76, Dept of Operations Research, Stanford Univ., Stanford, California (1977).

[59] Reiser, M. and Kobayashi, H., "Accuracy of the Diffusion Approximation for some Queueing Systems," *IBM J. of Res. Dev.*, 18, pp. 110-123 (1974).

[60] Schassburger, R. and Daduna, H., "The Time for a Round Trip in a Cycle of Exponential Queues," *JACM*, 30, pp. 146-150 (1983).

[61] Serfozo, R.F., "An Equivalence Between Continuous and Discrete Time Markov Decision Processes," *Operations Research*, 27, pp. 616-620 (1979).

[62] Shalmon, M. and Kaplan, M.A., "A Tandem Network of Queues with Deterministic Service and Intermediate Arrivals," *Operations Research*, 32, pp. 753-773 (1984).

[63] Shantikumar, J.G. and Buzacott, J.A., "Open Queueing Network Models of Dynamic Job Shops," *Int. J. Prod. Res.*, 19, pp. 255-266 (1981).

[64] Simon, B. and Foley, R.D., "Some Results on Sojourn Times in Acyclic Jackson Networks," *Management Science*, 25, pp. 1027-1034 (1979).

[65] Smith, R.J., *Circuits, Devices, and Systems*, John Wiley and Sons, New York (1984).

[66] Takacs, L., *Stochastic Processes*, John Wiley and Sons, Inc., New York (1962).

[67] Walrand, J. and Varaiya, P., "Sojourn Times and the Overtaking Condition in Jacksonian Networks," *Adv. Appl. Prob.*, 12, pp. 1000-1018 (1980).

[68] Wenocur, M.L., "A Production Network Model and its Diffusion Approximation," Tech Rep No.207, Dept of Operations Research, Stanford Univ., Stanford, California (1982).

[69] Whitt, W., *Heavy Traffic Limit Theorems for Queues: A Survey, Lecture Notes in Economics and Mathematical Sciences, No.98*, Springer-Verlag, New York (1973).

[70] Whittle, P., "Equilibrium Distributions for an Open Migration Process," *J. Appl. Prob.*, 5, pp. 567-571 (1968).

[71] Whittle, P., "Nonlinear Migration Processes," *Inter. Stat. Inst. Bull*, 42, Part 1, pp. 642-647 (1969).

[72] Wong, J.W., "Distribution of End-to-End Delay in Message- Switched Networks," *Comp. Networks*, 2, pp. 44-49 (1978).

BIOGRAPHICAL SKETCH

Russell Edward King was born March 14, 1958, in Fort Pierce, Florida. He received elementary and secondary education in Fort Pierce and Rockledge, Florida, graduating from Rockledge High School as valedictorian of the class of 1976. He attended the University of Florida from 1976 to 1980, receiving a bachelor's degree in systems engineering, with honors, in August 1980. In September 1980 he enrolled in the graduate program in industrial and systems engineering at the University of Florida. While pursuing a graduate degree he was employed as an industrial engineer by the Naval Air Rework Facility at the Jacksonville Naval Air Station, Jacksonville, Florida. He received his master's degree in industrial engineering in May, 1982.

Upon graduation he continued at the University of Florida to pursue his PhD. In July 1983 he was employed by Dynamac Corporation at their Lexington Park, Maryland, facility as a functional systems analyst, applying his dissertation research to the development of a revised Workload Control System for the United States Navy's Naval Air Rework Facilities. At the same time he began working for North Carolina State University as

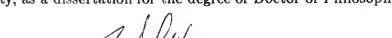
a research associate and moved to Raleigh, North Carolina, commuting to Maryland to work for Dynamac on alternating weeks. In July 1985 Russell King joined the faculty at North Carolina State University as an Assistant Professor pending the completion of his PhD. He received the Clifford A. Anderson Award as the Outstanding Faculty Member in Industrial Engineering for the academic year 1985.

Russell King is a member of the Tau Beta Pi and Alpha Pi Mu honorary engineering societies as well as the Institute of Industrial Engineers, The Institute of Management Science, and the Operations Research Society of America. He is married to the former Christina Clare Unrein of Satellite Beach, Florida.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Thom J. Hodgson, Chairman
Professor of Industrial and Systems Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

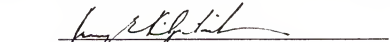


Louis A. Martin-Vega, Co-Chairman
Associate Professor of Industrial and Systems Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

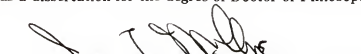
Ralph W. Swain
Professor of Industrial and Systems Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Kerry E. Kilpatrick
Professor of Industrial and Systems Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



James T. McClave
Associate Professor of Management

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December 1986

Herbert A. Beus

Dean, College of Engineering

Dean, Graduate School